

**DIAGNOSIS HAMA PENYAKIT TANAMAN BAWANG MERAH  
MENGUNAKAN ALGORITMA *MODIFIED K-NEAREST  
NEIGHBOR* (MKNN)**

**SKRIPSI**

Untuk memenuhi sebagian persyaratan  
memperoleh gelar Sarjana Komputer  
Keminatan : Komputasi Cerdas

Disusun oleh:

Mohamad Yusuf Arrahman

NIM: 135150201111132



PROGRAM STUDI TEKNIK INFORMATIKA  
JURUSAN TEKNIK INFORMATIKA  
FAKULTAS ILMU KOMPUTER  
UNIVERSITAS BRAWIJAYA  
MALANG  
2018

## PENGESAHAN

### PENGESAHAN

DIAGNOSIS HAMA PENYAKIT TANAMAN BAWANG MERAH MENGGUNAKAN  
ALGORITMA MODIFIED K-NEAREST NEIGHBOR (MKNN)

SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan  
memperoleh gelar Sarjana Komputer  
Keminatan : Komputasi Cerdas

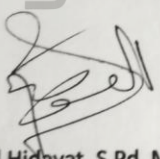
Disusun Oleh :  
Mohamad Yusuf Arrahman  
NIM: 135150201111132


Skripsi ini telah diuji dan dinyatakan lulus pada  
27 Juli 2018

Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I

Dosen Pembimbing II


  
Nurul Hidayat, S.Pd, M.Sc  
NIP:196804302002121001

  
Ir. Sutrisno, M.T  
NIP. 195703251987011001

Mengetahui

Ketua Jurusan Teknik Informatika



  
Tri Astoto Kurniawan, S.T, M.T, Ph.D  
NIP: 197105182003121001

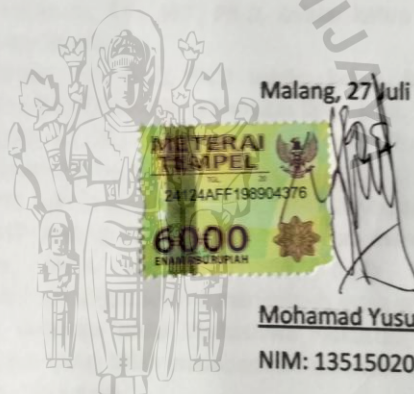
## PERNYATAAN ORISINALITAS

### PERNYATAAN ORISINALITAS

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya, di dalam naskah skripsi ini tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu perguruan tinggi, dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis disitasi dalam naskah ini dan disebutkan dalam daftar pustaka.

Apabila ternyata didalam naskah skripsi ini dapat dibuktikan terdapat unsur-unsur plagiasi, saya bersedia skripsi ini digugurkan dan gelar akademik yang telah saya peroleh (sarjana) dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku (UU No. 20 Tahun 2003, Pasal 25 ayat 2 dan Pasal 70).

Malang, 27 Juli 2018



Mohamad Yusuf Arrahman

NIM: 135150201111132

## KATA PENGANTAR

Puji syukur kehadiran Tuhan YME yang telah melimpahkan rahmat-Nya sehingga laporan skripsi yang berjudul “Diagnosis Hama Penyakit Tanaman Bawang Merah Menggunakan Algoritma Modified K-Nearest Neighbor (MKNN)” ini dapat terselesaikan.

Penulis menyadari bahwa skripsi ini tidak akan berhasil tanpa bantuan dari beberapa pihak. Oleh karena itu, penulis ingin menyampaikan rasa hormat dan terimakasih kepada:

1. Nurul Hidayat, S.Pd, M.Sc dan Ir. Sutrisno M.T selaku dosen pembimbing skripsi yang telah dengan sabar membimbing dan mengarahkan penulis sehingga dapat menyelesaikan skripsi ini.
2. Wayan Firdaus Mahmudy, S.Si, M.T, Ph.D, selaku ketua Fakultas Ilmu Computer Universitas Brawijaya.
3. Tri Astoto Kurniawan, S.T, M.T, Ph.D selaku ketua Jurusan Teknik Informatika Fakultas Ilmu Komputer Universitas Brawijaya.
4. Agus Wahyu Widodo, S.T, M.Cs selaku ketua Program Studi Informatika Fakultas Ilmu Komputer Universitas Brawijaya.
5. Djoko Joedomiatmo dan Tumini Sriwahyuti beserta keluarga besar atas segala nasehat, kasih sayang, dan kesabaran di dalam membesarkan dan mendidik penulis, serta yang senantiasa tiada henti-hentinya memberikan doa dan semangat selama menempuh masa studi.
6. Sahabat sekaligus keluarga besar mahasiswa Fakultas Ilmu Komputer Universitas Brawijaya yang telah memberikan motivasi dan dukungan selama menempuh masa studi.
7. Seluruh civitas akademika Informatika Universitas Brawijaya yang telah banyak memberi bantuan selama penulis menempuh studi di Informatika Universitas Brawijaya

Penulis menyadari bahwa dalam penyusunan skripsi ini masih banyak kekurangan, sehingga saran dan kritik yang membangun sangat penulis harapkan. Akhir kata penulis berharap skripsi ini dapat membawa manfaat bagi semua pihak.

Malang, 27 Juli 2018

Penulis

Mohamad Yusuf Arrahman

Yusuf.saymeanst@gmail.com

## ABSTRAK

Tanaman bawang merah (*Allium cepa* L.) merupakan sayuran rempah yang cukup populer di Indonesia, memiliki nilai ekonomis tinggi, berfungsi sebagai penyedap rasa, dan dapat digunakan sebagai bahan obat tradisional. Akan tetapi kendala dijumpai dalam proses penanaman bawang merah, salah satunya hama dan penyakit yang sering mengakibatkan gagal panen. Salah satu metode untuk mendiagnosis penyakit tanaman bawang merah bisa dilakukan dengan *modified k-nearest neighbor* (MKNN). Sistem pakar diagnosis penyakit tanaman bawang merah menggunakan metode *modified k-nearest neighbor* (MKNN) ini dapat mempermudah untuk mendeteksi penyakit yang menyerang bawang merah berdasarkan gejala –gejala yang timbul kemudian dari gejala tersebut diproses dengan metode MKNN dengan hasil akhir diagnosis salah satu hama penyakit dari bawang merah. Metode *modified k-nearest neighbor* (MKNN) diimplementasikan pada mesin inferensi sistem pakar agar dapat melakukan penarikan kesimpulan berdasarkan pengetahuan yang ada pada basis pengetahuan. Hasil yang didapatkan setelah pengujian akurasi sistem yaitu 83,33% yang menunjukkan bahwa metode *modified k-nearest neighbor* (MKNN) cocok untuk kasus penyakit tanaman bawang merah.

Kata kunci: Sistem Pakar, *Modified K-Nearest Neighbor*, Hama Penyakit Bawang Merah.



## ABSTRACT

*Red onion (*Allium cepa* L.) is a spice vegetable that is quite popular in Indonesia, has high economic value, serves as flavoring, and can be used as a traditional medicine ingredient. However, obstacles encountered in the process of planting onions, one of the pests and diseases that often lead to crop failure. One method to diagnose diseases of shallot plants can be done with modified k-nearest neighbor (MKNN). The expert system of onion plant disease diagnosis using the k-nearest neighbor (MKNN) modified method can make it easier to detect diseases that attack onion based on later symptoms of the symptoms processed by the MKNN method with the final diagnosis of one pest of disease red onion. The k-nearest neighbor (MKNN) modified method is implemented on an expert system inference engine in order to draw conclusions based on existing knowledge on the knowledge base. The results obtained after the system accuracy test of 83.33% indicating that the modified method k-nearest neighbor (MKNN) is suitable for cases of onion plant diseases*

**Keywords:** Expert System, Modified K-Nearest Neighbor, Pest Disease Onion.



## DAFTAR ISI

PENGESAHAN .....	ii
PERNYATAAN ORISINALITAS .....	iii
KATA PENGANTAR.....	iv
ABSTRAK.....	v
ABSTRACT .....	vi
DAFTAR ISI .....	vii
DAFTAR TABEL.....	x
DAFTAR GAMBAR.....	xi
DAFTAR KODE PROGRAM .....	xii
DAFTAR LAMPIRAN .....	xiii
BAB 1 PENDAHULUAN.....	1
1.1 Latar belakang.....	2
1.2 Rumusan masalah.....	3
1.3 Tujuan .....	3
1.4 Manfaat.....	3
1.5 Batasan masalah .....	3
1.6 Sistematika pembahasan.....	4
BAB 2 LANDASAN KEPUSTAKAAN .....	6
2.1 Penelitian Terkait.....	6
2.2 Data Mining.....	7
2.2.1 Definisi Data Mining.....	7
2.2.2 Tahap-Tahap Data Mining.....	7
2.2.3 Metode Data Mining.....	8
2.3 Klasifikasi.....	8
2.4 K-Nearest Neighbor (KNN).....	9
2.4.1 Definisi K-Nearest Neighbor (KNN) .....	9
2.4.2 Proses K-Nearest Neighbor (KNN) .....	10
2.5 Modified K-Nearest Neighbor (MKNN).....	11
2.5.1 Definisi Modified K-Nearest Neighbor (MKNN) .....	11
2.5.2 Proses Modified K-Nearest Neighbor (MKNN) .....	11

2.6 Tanaman Bawang Merah .....	13
2.7 Hama Penyakit Tanaman Bawang Merah.....	13
2.7.1 Hama Ulat Bawang ( <i>Spodoptera Exigua</i> ) .....	13
2.7.2 Hama Lalat Pengorok Daun ( <i>Lirionzoa chinensis</i> ) .....	13
2.7.3 Penyakit Moler atau Layu <i>Fusarium</i> ( <i>Twisting Disease</i> ) .....	14
2.7.4 Penyakit Trotol atau Bercak Ungu ( <i>Alternaria Porri</i> ) .....	14
2.7.5 Penyakit Otomatis atau <i>Antraknose</i> ( <i>Antracnose</i> ) .....	14
2.8 Akurasi Sistem.....	14
BAB 3 METODOLOGI .....	16
3.1 Studi literatur .....	16
3.2 Pengumpulan data .....	17
3.3 Analisis kebutuhan.....	17
3.4 Analisis dan perancangan sistem.....	18
3.5 Implementasi .....	18
3.6 Pengujian Sistem.....	18
3.7 Evaluasi .....	19
BAB 4 Perancangan .....	20
4.1 Deskripsi sistem .....	20
4.2 Preproses data .....	21
4.3 Perancangan perangkat lunak .....	23
4.3.1 Perancangan proses .....	23
4.4 Perhitungan manual .....	30
BAB 5 Implementasi .....	37
5.1 Implementasi sistem.....	37
5.1.1 Spesifikasi perangkat keras .....	37
5.1.2 Spesifikasi Perangkat Lunak .....	38
5.2 Batasan implementasi .....	38
5.3 Implementasi algoritma.....	38
5.3.1 Implementasi algoritma perhitungan normalisasi data.....	39
5.3.2 Implementasi algoritma perhitungan <i>euclidean</i> .....	40
5.3.3 Implementasi algoritma perhitungan validitas data.....	40
5.3.4 Implementasi algoritma perhitungan <i>weight voting</i> .....	42



5.4 Implementasi antarmuka.....	44
5.4.1 Tampilan antarmuka utama .....	44
5.4.2 Tampilan antarmuka pemilihan gejala dan nilai k.....	45
5.4.3 Tampilan antarmuka hasil diagnosis .....	46
BAB 6 Pengujian Dan Analisis.....	48
BAB 7 Penutup .....	53
7.1 Kesimpulan.....	53
7.2 Saran .....	53
DAFTAR PUSTAKA.....	54

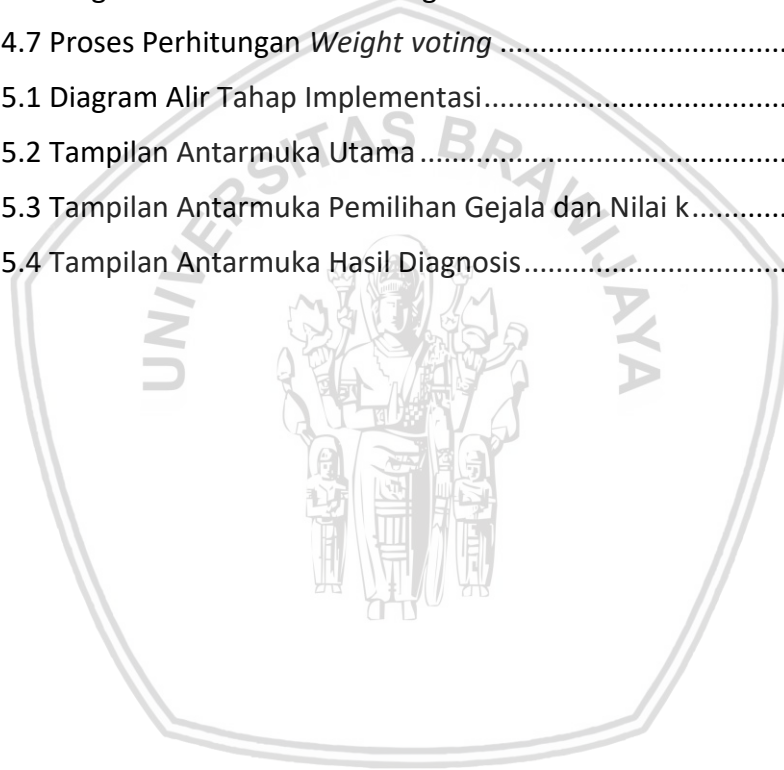


## DAFTAR TABEL

Tabel 4.1 Nilai Pembobotan 15 Gejala Hama Penyakit Tanaman Bawang Merah Dari Pakar .....	21
Tabel 4.2 Pasangan Gejala dan Penyakit yang Menyerang Bawang Merah .....	23
Tabel 4.3 Dataset Perhitungan Manual .....	30
Tabel 4.4 Data Latih .....	31
Tabel 4.5 Data Uji .....	31
Tabel 4.6 Data Latih Hasil Normalisasi .....	32
Tabel 4.7 Data Uji Hasil Normalisasi .....	32
Tabel 4.8 Perhitungan <i>Euclidean</i> Data Latih .....	33
Tabel 4.9 Perhitungan Validitas .....	34
Tabel 4.10 Tabel Perhitungan <i>Euclidean</i> Data Uji 1 .....	34
Tabel 4.11 Tabel Nilai <i>Weight Voting</i> Data Uji 1 dengan 3 Nilai Terbesar .....	35
Tabel 4.12 Hasil Perhitungan 5 Data Uji .....	36
Tabel 5.1 Spesifikasi Perangkat Keras .....	37
Tabel 5.2 Spesifikasi Perangkat Lunak .....	38
Tabel 6.1 Pengaruh Nilai k .....	48

## DAFTAR GAMBAR

Gambar 3.1 Diagram Alir Metodologi Penelitian.....	16
Gambar 4.1 Diagram Alir Tahap Perancangan Sistem .....	20
Gambar 4.2 Diagram Alir Sistem .....	24
Gambar 4.3 Diagram Alir Proses Perhitungan Normalisasi .....	25
Gambar 4.4 Diagram Alir Proses Algoritma MKNN.....	26
Gambar 4.5 Diagram Alir Proses Perhitungan Validitas .....	28
Gambar 4.6 Diagram Alir Proses Perhitungan Jarak <i>Euclidean</i> .....	29
Gambar 4.7 Proses Perhitungan <i>Weight voting</i> .....	30
Gambar 5.1 Diagram Alir Tahap Implementasi.....	37
Gambar 5.2 Tampilan Antarmuka Utama .....	45
Gambar 5.3 Tampilan Antarmuka Pemilihan Gejala dan Nilai k.....	46
Gambar 5.4 Tampilan Antarmuka Hasil Diagnosis.....	47



DAFTAR KODE PROGRAM

Source Code 5.1 *Listing Code* Proses Normalisasi Data ..... 39

Source Code 5.2 *Listing Code* Proses Euclidean ..... 40

Source Code 5.3 *Listing Code* Proses Validitas..... 42

Source Code 5.4 *Listing Code* Proses Weight Voting ..... 44



## DAFTAR LAMPIRAN

LAMPIRAN A HASIL WAWANCARA DENGAN PAKAR .....	56
LAMPIRAN B NILAI PEMBOBOTAN GEJALA HAMA PENYAKIT BAWANG MERAH .	57
LAMPIRAN C DATA LATIH SISTEM .....	59
LAMPIRAN D DATA KASUS UJI.....	61



## BAB 1 PENDAHULUAN

Indonesia merupakan negara agraris karena sebagian besar atau mayoritas penduduknya bermata pencaharian dengan cara bertani, banyak tanaman yang dapat hidup di Indonesia salah satunya adalah tanaman bawang merah. Bawang merah merupakan sejenis tanaman yang dapat dijadikan bumbu berbagai masakan di dunia termasuk di Indonesia. Akan tetapi kendala dijumpai dalam proses penanaman bawang merah, salah satunya hama dan penyakit yang sering mengakibatkan gagal panen. Untuk mencegah dan menanggulangi hama dan penyakit tersebut para petani tentunya membutuhkan solusi dari seorang yang benar benar ahli dalam bidang penanaman bawang merah untuk memberikan solusi dan langkah untuk mengatasi permasalahan tersebut sehingga resiko gagal panen dapat dihindari. Oleh karena itu, dibutuhkan suatu alat bantu yang dapat mendiagnosis jenis hama dan penyakit pada tanaman bawang merah berupa suatu sistem pakar. Sistem pakar tersebut dapat dimanfaatkan oleh petani sebagai alat bantu untuk mengambil sebuah keputusan dan mengetahui jenis hama dan penyakit tersebut.

Penerapan sistem pakar ini dapat dijalankan melalui aplikasi yang berbasis android yang saat ini sangat populer. Selain untuk petani sistem pakar ini juga bermanfaat bagi pengamat hama penyakit sebagai alat bantu untuk mengambil keputusan tentang jenis hama atau penyakit. Tujuan dari penelitian ini yaitu membuat perangkat lunak yang *user friendly* dan dapat digunakan oleh masyarakat luas serta dapat sebagai pembelajaran atau sosialisasi hama dan penyakit pada tanaman bawang merah sekaligus memberikan solusi penanganannya.

Hama dan penyakit pada tanaman bawang merah secara berkala sering kurang diperhatikan sehingga membuat tanaman bawang merah sering kurang optimal hasil panennya, bahkan bisa sampai gagal panen. Para petani dan penyuluh juga banyak yang mengeluhkan banyaknya hama dan penyakit yang menyerang tanaman bawang merah. Untuk itu diperlukan sebuah sistem pakar terhadap hama dan penyakit pada tanaman bawang merah secara tepat dan akurat, dikarenakan hama dan penyakit pada tanaman tersebut dapat dengan mudah menyebar serta menyerang keseluruhan lahan pertanian.

Dalam hal ini peran seorang pakar sangat dibutuhkan dalam mendiagnosis jenis hama dan penyakit serta memberikan cara penanggulangannya untuk mendapatkan solusi terbaik. Demikian pula jika ditemukan adanya jenis hama dan penyakit baru pada tanaman tersebut, maka seorang expert harus melakukan penelitian untuk mendapatkan keterangan-keterangan dari hama atau penyakit baru tersebut dan secepat mungkin memberikan sosialisasi kepada para petani mengenai jenis hama dan penyakit baru tersebut beserta cara penanganannya. Namun demikian, keterbatasan yang dimiliki seorang expert terkadang menjadi kendala bagi para petani yang akan melakukan konsultasi untuk menyelesaikan suatu



permasalahan dan mendapatkan solusi terbaik. Dalam hal ini sistem pakar dihadirkan sebagai alternatif kedua dalam memecahkan permasalahan setelah seorang expert.

## 1.1 Latar belakang

Untuk mengetahui jenis hama penyakit tanaman bawang merah yang semula harus datang ke pakar atau ahli tanaman bawang merah yang membutuhkan banyak waktu dan relatif lama dapat diperbaiki dengan aplikasi android. Aplikasi android yang dapat digunakan yaitu membuat suatu sistem untuk mendiagnosis hama penyakit tanaman bawang merah dengan menggunakan metode tertentu. Dengan sistem ini informasi dapat disampaikan jauh lebih cepat dan mudah dimengerti.

Metode *K-Nearest Neighbor* merupakan salah satu metode klasifikasi yang sederhana dan banyak digunakan. Metode KNN (*K – Nearest Neighbor*) adalah suatu metode yang menggunakan klasifikasi ketetanggaan yang bertujuan untuk mengklasifikasikan objek baru berdasarkan atribut (*feature*) dan *training sample*. Kelebihan metode KNN ini yaitu lebih efektif di data latih dengan jumlah besar dan dapat menghasilkan data yang lebih akurat. Sedangkan untuk kelemahan Metode KNN ini yaitu terletak pada rendahnya tingkat akurasi pada dataset multidimensional dapat diatasi dengan melakukan modifikasi terhadap metode KNN (Parvin., et al., 2010).

Metode *Modified K-Nearest Neighbor* (MKNN) merupakan suatu metode yang mengklasifikasikan objek baru dengan cara menempatkan label kelas data sesuai dengan nilai k yang divalidasi. Nilai k dan besarnya data latih berpengaruh dalam penggunaan metode klasifikasi MKNN (Zainuddin, et al., 2013).

Pemecahan masalah untuk mendiagnosis hama penyakit pada tanaman bawang merah pernah diusulkan pada penelitian sebelumnya. Penelitian pertama membahas tentang sistem pakar untuk mendiagnosis hama penyakit pada tanaman bawang merah. Sistem pakar yang dikembangkan pada penelitian ini berbasis web dan menggunakan metode *dempster-shafer* (T.A. Gunawan, et al., 2016).

Penelitian lainnya membahas tentang penerapan metode M-KNN untuk mengklasifikasikan penyakit tanaman tomat. Pada penelitian ini menghasilkan tingkat akurasi rata-rata maksimum sebesar 98,92% (Prasiwi M.B., et al., 2015).

Berdasarkan permasalahan tersebut maka penulis menggunakan metode *Modified K-Nearest Neighbor* untuk diagnosis hama penyakit tanaman bawang merah. Input dari sistem ini berupa gejala-gejala yang terdapat pada tanaman bawang merah, sedangkan untuk outputnya berupa hasil diagnosis hama penyakit dan cara pengendaliannya. Harapan dari hasil implementasi Metode *Modified K-Nearest Neighbor* untuk diagnosis hama penyakit tanaman bawang merah ini dapat membantu petani bawang merah lebih

cepat dan tanggap dalam mengidentifikasi jenis penyakit yang menyerang tanamannya serta cara pengendaliannya.

## 1.2 Rumusan masalah

Berdasarkan uraian dari latar belakang tersebut, maka rumusan masalah adalah sebagai berikut :

1. Bagaimana mengimplementasikan metode *Modified K-Nearest Neighbor* untuk mendiagnosis penyakit tanaman bawang merah.
2. Bagaimana hasil akurasi dari diagnosis penyakit tanaman bawang merah menggunakan metode *Modified K-Nearest Neighbor*.

## 1.3 Tujuan

Tujuan dari penelitian ini adalah sebagai berikut:

1. Mengimplementasikan metode *Modified K-Nearest Neighbor* ke dalam program aplikasi untuk mendiagnosis penyakit tanaman bawang merah.
2. Untuk mengetahui hasil akurasi dari implementasi metode *Modified K-Nearest Neighbor* pada proses diagnosis penyakit tanaman bawang merah.

## 1.4 Manfaat

Manfaat dari penelitian ini adalah sebagai berikut:

1. Memperoleh informasi terkait penyakit tanaman bawang merah serta gejala-gejalanya.
2. Mempermudah para petani bawang merah dalam mengidentifikasi hama penyakit yang ada pada bawang merah serta cara menanggulangi.
3. Mengetahui seberapa akurat hasil metode *Modified K-Nearest Neighbor* terhadap permasalahan ini.

## 1.5 Batasan masalah

Agar tidak terlepas dari maksud dan tujuan dalam penyusunan laporan maka peneliti membatasi pokok permasalahan pada:

1. Mendiagnosis jenis penyakit dan hama pada tanaman bawang merah dengan melihat gejala yang berikutnya akan muncul keterangan serta solusi yang harus dilakukan.
2. Data akan di ambil dari Balai Pengkajian Teknologi Pertanian (BPTP) Jawa Timur.
3. Data terdiri dari 5 hama penyakit dan 15 Gejala.

4. Proses Diagnosis berdasarkan gejala yang tampak pada umbi, daun dan tanaman bawang merah.
5. Sistem yang dirancang hanya untuk mendiagnosis penyakit tanaman bawang merah dan cara pengendaliannya.
6. Metode yang digunakan pada sistem ini yaitu metode *Modified K-Nearest Neighbor*.

## 1.6 Sistematika pembahasan

Untuk mencapai tujuan yang diharapkan, maka sistematika penulisan yang disusun dalam tugas akhir ini adalah sebagai berikut:

### BAB I Pendahuluan

Bab ini berisi tentang latar belakang, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, dan sistematika penulisan.

### BAB II Tinjauan Pustaka

Tinjauan Pustaka menjelaskan tentang kajian pustaka terkait dengan penelitian yang telah ada seperti penelitian tentang “Implementasi algoritma *Modified K-Nearest Neighbor*(MKNN) untuk menentukan penyakit pada tanaman tomat (Prasiwi M.B., et al., 2015)”. Dasar teori yang diperlukan untuk mendukung penelitian adalah sistem pakar, Metode *K- Nearest Neighbor*, Metode *Modified K-Nearest Neighbor*, Pengertian tanaman bawang merah, jenis hama penyakit tanaman bawang merah, dan akurasi sistem.

### BAB III Metodologi

Membahas tentang metode yang digunakan dalam penulisan yang terdiri dari studi literatur, pengambilan data sampel dari penyakit tanaman bawang merah, Analisis kebutuhan, Analisis dan perancangan sistem, implementasi, uji coba sistem dan pengambilan kesimpulan dari aplikasi untuk Diagnosis penyakit pada tanaman bawang merah menggunakan metode *Modified K-Nearest Neighbor*.

### BAB IV Analisis Kebutuhan dan Perancangan

Membahas tentang Analisis kebutuhan dari pemodelan sistem pakar untuk Diagnosis hama penyakit pada tanaman bawang merah menggunakan metode *Modified K- Nearest Neighbor*.

### BAB V Implementasi

Membahas tentang hasil perancangan dari analisis kebutuhan dan implementasi dari aplikasi Diagnosis hama penyakit pada tanaman bawang merah menggunakan metode *Modified K-Nearest Neighbor*.

### BAB VI Pengujian dan Analisis

Memuat tentang hasil pengujian dan analisis terhadap implementasi dari aplikasi untuk Diagnosis hama penyakit pada tanaman bawang merah menggunakan metode *Modified K-Nearest Neighbor* yang telah direalisasikan.

#### BAB VII Penutup

Memuat kesimpulan yang diperoleh dari pembuatan dan pengujian aplikasi untuk Diagnosis hama penyakit pada tanaman bawang merah menggunakan metode *Modified K- Nearest Neighbor* yang dikembangkan dalam skripsi ini serta saran-saran untuk pengembangan lebih lanjut.



## BAB 2 LANDASAN KEPUSTAKAAN

Pada bab dua, terdiri dari penelitian terkait dan dasar teori. Penelitian terkait membahas penelitian yang telah ada sebelumnya. Dasar teori membahas teori yang diperlukan untuk menyusun penelitian yang diusulkan. Kajian pustaka pada penelitian ini merupakan penelitian yang sudah ada sebelumnya yang berjudul “Sistem Pakar Diagnosis Penyakit Tanaman Tomat Menggunakan Algoritma *Modified K-Nearest Neighbor*” dan “Sistem Pakar Mendiagnosis Hama Penyakit Bawang Merah Menggunakan Metode *Dempster-Shafer*”. Dasar teori yang diperlukan berdasarkan latar belakang dan rumusan masalah adalah konsep dasar *Data mining*, metode K-NN, metode *Modified K-Nearest Neighbor* dan penjelasan tentang hama penyakit tanaman bawang merah.

### 2.1 Penelitian Terkait

Pada sub bab ini dilakukan Analisis perbandingan terhadap beberapa penelitian sebelumnya terkait diagnosis hama penyakit tanaman bawang merah. Selain itu, juga akan dibahas penelitian terkait tentang penggunaan metode *Modified K-Nearest Neighbor* yang juga diusulkan pada penelitian penulis.

Penelitian pertama adalah penelitian yang dilakukan oleh Teguh A.G. dkk. Penelitian ini berjudul “Pemodelan Sistem Pakar Diagnosis Hama Penyakit Bawang Merah Dengan Metode *Dempster-Shafer*”, Hama penyakit tanaman bawang merah dijadikan sebagai masalah utama dalam penelitian dan dikembangkan berbasis aplikasi web. Hama penyakit dalam penelitian tersebut berjumlah 5 dengan masukan sistem berupa 23 gejala yang terjadi pada bawang merah. Berdasarkan data tersebut, sistem ini mampu menghasilkan tingkat keakurasian sebesar 93,54%(Teguh A.G., et al., 2016).

Penelitian kedua adalah penelitian yang dilakukan oleh Prasiwi M.B. dkk. Penelitian ini berjudul “Diagnosis Penyakit tanaman Tomat menggunakan Algoritma *Modified K-Nearest Neighbor*(MKNN)”, pada penelitian menjelaskan tentang penerapan metode *Modified K-Nearest Neighbor* untuk mendiagnosis penyakit tanaman tomat dengan menggunakan data morfologi tanaman tomat. Data yang digunakan terdiri dari 6 jenis penyakit dan 15 gejala. Pada penelitian ini rata-rata tingkat akurasi maksimum sebesar 98,92%.(Prasiwi M.B., et al., 2015)

Perbedaan pada penelitian sebelumnya dan penelitian ini adalah data masukan yang digunakan pada penelitian ini berupa gejala-gejala dari penyakit tanaman bawang merah yang terdiri dari 5 hama penyakit dan 15 gejala hama penyakit yang diambil pada 2016 oleh Teguh A.G. di Balai Pengkajian Teknologi Pertanian (BPTP) Malang Jawa Timur. Selanjutnya proses perhitungan yang digunakan yaitu dengan metode *Modified K-*

*Nerarest Neighbor*(MKNN). Sedangkan keluaran dalam sistem ini berupa diagnosis hama penyakit tanaman bawang merah dan cara pengendaliannya.

## 2.2 Data Mining

Pada sub bab ini akan dijelaskan mengenai definisi dari *data mining*, tahap- tahap dalam *data mining* dan metode yang terdapat pada *data mining*.

### 2.2.1 Definisi Data Mining

*Data mining* didefinisikan sebagai satu set teknik yang digunakan secara otomatis untuk mengeksplorasi secara menyeluruh dan membawa ke permukaan relasi-relasi yang kompleks pada suatu set data yang besar (Moertini, 2002).

Menurut Larose, Data mining yaitu bidang yang digabung dari beberapa bidang keilmuan yang menyatukan teknik dari pembelajaran mesin, pengenalan pola, statistik, *database* dan visualisasi untuk pengenalan permasalahan pengambilan informasi dari *database* yang benar (Larose, 2005).

Ada beberapa definisi *Data mining* antara lain (Kusnawi, 2007):

1. *Data mining* yaitu Analisis otomatis dari suatu data yang jumlahnya sangat besar atau kompleks dengan tujuan untuk menemukan pola atau kecenderungan penting yang biasanya tidak disadari keberadaannya.
2. *Data mining* yaitu serangkaian proses untuk menggali nilai tambah dari suatu kumpulan data berupa pengetahuan yang selama ini tidak diketahui secara manual.
3. *Data mining* atau *Knowledge Discovery in Databases* (KDD) yaitu pengambilan informasi yang tersembunyi, dimana informasi tersebut sebelumnya tidak dikenal dan berpotensi bermanfaat. Proses ini meliputi sejumlah pendekatan teknis yang berbeda, seperti *clustering*, *data summarization* dan *learning classification rules*.

### 2.2.2 Tahap-Tahap Data Mining

Dalam *Data mining* terdapat beberapa tahap-tahap yang dilakukan untuk menemukan suatu model dari data, diantaranya (Kantardzic, 2003):

1. Merumuskan permasalahan : Dalam tahapan perumusan masalah ini ditetapkan rumusan masalah dan variabel-variabel yang terlibat.
2. Pengumpulan data : Pada tahapan ini dilakukan proses pembuatan data dan pengumpulan data yang akan digunakan.



3. *Preprocessing data*: Dalam tahapan ini dilakukan seleksi data yang akan digunakan dalam proses.
4. Estimasi model seleksi dan implementasi : Pada tahapan ini dilakukan estimasi model dan implementasi terhadap metode *data mining* yang tepat.
5. Menafsirkan model dan menarik kesimpulan : Pada tahapan terakhir dilakukan penafsiran model dari *data mining* dan ditarik kesimpulan dari hasil akhir yang didapat.

### 2.2.3 Metode Data Mining

Metode pada *Data mining* dapat dikelompokkan menjadi 2 kategori utama, yaitu:

1. *Descriptive mining*

*Descriptive mining* yaitu suatu proses untuk menemukan karakteristik penting dari data di suatu basis data. Teknik *Data mining* yang termasuk dalam *descriptive mining* adalah clustering, *sequential mining* dan *association*.

2. *Predictive mining*

data dengan menggunakan beberapa variabel lain di masa depan. Salah satu teknik yang terdapat dalam *predictive mining* yaitu metode klasifikasi.

## 2.3 Klasifikasi

Klasifikasi adalah salah satu metode dari *Data mining* untuk menemukan model atau fungsi yang menjelaskan atau membedakan konsep atau kelas data, dengan tujuan untuk dapat memperkirakan kelas dari suatu objek yang labelnya tidak diketahui (Pramudiono, 2003). Klasifikasi bertujuan untuk memprediksi kelas dari suatu objek yang labelnya tidak diketahui.

Selain itu klasifikasi juga dapat berarti suatu teknik dengan melihat pada kelakuan dan atribut kelompok yang telah didefinisikan. Teknik tersebut dapat memberikan klasifikasi data baru dengan memanipulasi data yang telah diklasifikasikan sebelumnya dan hasil tersebut digunakan untuk memberikan sejumlah aturan-aturan pada data baru untuk diklasifikasikan. Teknik ini menggunakan *supervised induction*, yang memanfaatkan kumpulan pengujian dari *record* yang terklasifikasi untuk menentukan kelas-kelas tambahan (Kusnawi, 2007).

Secara umum, proses klasifikasi biasanya dibagi menjadi dua fase, yaitu (Pramudiono, 2003):

#### 1. Learning

Pada fase ini, sebagian data yang telah diketahui kelas datanya diumpankan untuk membentuk model perkiraan.

#### 2. Test

Pada fase test, model yang sudah terbentuk diuji dengan sebagian data lainnya untuk mengetahui akurasi dari model tersebut. Bila kurasinya mencukupi maka model ini dapat digunakan untuk memprediksi kelas data yang belum diketahui.

Pada proses klasifikasi terdapat beberapa metode, antara lain *decision tree*, *Bayesian*, *fuzzy*, *neural network*, *support vector machine* (SVM) dan *k-nearest neighbor* (Pramudiono, 2003).

### 2.4 K-Nearest Neighbor (KNN)

Pada sub bab ini akan dijelaskan mengenai definisi dari *K-Nearest Neighbor* (KNN), proses dari (KNN) dan tahapan dari proses KNN.

#### 2.4.1 Definisi K-Nearest Neighbor (KNN)

*K-Nearest Neighbor* (KNN) merupakan suatu metode yang biasa digunakan untuk klasifikasi data. Algoritma KNN adalah sebuah metode untuk mengklasifikasikan objek berdasarkan data pembelajaran yang jaraknya paling dekat dengan objek tersebut. KNN adalah suatu metode yang menggunakan algoritma *supervised* dengan hasil dari *query instance* yang baru diklasifikasikan berdasarkan mayoritas dari kategori pada KNN. Tujuan dari algoritma ini yaitu mengklasifikasikan objek baru berdasarkan atribut dan *training sample* (Aktivia, 2012).

Prinsip umum dari algoritma ini adalah menemukan k data training untuk menentukan *K-Nearest Neighbor* berdasarkan ukuran jarak. Selanjutnya mayoritas dari k tetangga terdekat akan menjadi dasar untuk memutuskan kategori dari sample berikutnya (Yu, 2010).

Metode KNN merupakan metode yang sederhana, mudah diimplementasikan, dan efektif jika data training besar. Meskipun memiliki beberapa kelebihan, metode KNN juga memiliki beberapa kelemahan seperti berikut (Parvin, et al., 2010):

1. Menggunakan perhitungan jarak yang belum diketahui pasti fungsi jarak yang digunakan.
2. Biaya komputasi yang tinggi karena perlu menghitung jarak setiap data training.
3. Perlu menentukan nilai k parameter, jumlah tetangga terdekat.

4. Membutuhkan memori yang besar.
5. Belum diketahui atribut yang lebih baik untuk menghasilkan hasil terbaik.
6. Rendahnya tingkat akurasi pada dataset multidimensi.

## 2.4.2 Proses K-Nearest Neighbor (KNN)

Prinsip kerja dari algoritma *K-Nearest Neighbor* yaitu membandingkan data uji yang diberikan dengan *data training* yang mirip dan mencari jarak terdekat antara data yang dievaluasi dengan k dari tetangga terdekat dalam *data training*. Ukuran dekat atau jauhnya tetangga biasanya dihitung berdasarkan variabel (Han & Kamber, 2006).

### 2.4.2.1 Interval Scaled Variable

*Interval scaled variable* merupakan ukuran-ukuran kontinu dari skala linier. Ukuran-ukuran tersebut berupa ukuran jarak, yang umum digunakan adalah jarak *Euclidean*. Perhitungan untuk menghitung jarak *Euclidean* dengan persamaan berikut (Han & Kamber, 2006):

$$d(X_1, X_2) = \sqrt{\sum_{i=1}^n (X_{1i} - X_{2i})^2} \quad (2.1)$$

Dimana  $X_1$  merupakan nilai variabel data testing dan  $X_2$  merupakan nilai variabel dari data training. Data kontinu dapat digunakan rumusan normalisasi atau standardisasi sebelum melakukan klasifikasi. Normalisasi bertujuan untuk mencegah atribut yang memiliki rentang terlalu besar dengan atribut yang bernilai kecil. Perhitungan min-max normalisasi dapat digunakan untuk mengubah atribut A dengan nilai v menjadi  $v'$  dalam range  $[0,1]$ . Perhitungan min-max normalisasi dengan persamaan (Han & Kamber, 2006):

$$v' = \frac{v - \min_A}{\max_A - \min_A} \times (\text{new}_{\max A} - \text{new}_{\min A}) + \text{new}_{\min A} \quad (2.2)$$

Dimana :

$v'$  = nilai normalisasi

$v$  = nilai data yang akan dinormalisasi

$\min_A$  = nilai terendah (minimal) data pada atribut A

$\max_A$  = nilai tertinggi (maximal) data pada atribut A

$\text{new}_{\min A}$  = nilai minimum terbaru yang diinginkan

$\text{new}_{\max A}$  = nilai maximum terbaru yang diinginkan

### 2.4.2.2 Tahapan Proses K-Nearest Neighbor (KNN)

Tahapan proses algoritma *K-Nearest Neighbor* (KNN) diawali dari menentukan parameter nilai  $k$  yang akan digunakan. Kemudian menghitung jarak antar data yang akan diuji dengan semua data latih menggunakan perhitungan jarak *euclidean*. Hasil dari perhitungan jarak *euclidean* tersebut diurutkan dari jarak terkecil ke jarak terbesar dan menentukan jarak terdekat sesuai dengan  $k$  yang digunakan. Dari jarak terdekat sesuai dengan  $k$  yang digunakan dipasangkan kelasnya kemudian kelas dari tetangga yang terdekat dicari jumlahnya dan ditetapkan kelas tersebut sebagai kelas data uji.

## 2.5 Modified K-Nearest Neighbor (MKNN)

Pada sub bab ini akan dijelaskan mengenai definisi dari metode *Modified K-Nearest Neighbor* dan proses dari metode *Modified K-Nearest Neighbor*.

### 2.5.1 Definisi Modified K-Nearest Neighbor (MKNN)

Algoritma *Modified K-Nearest Neighbor* (MKNN) adalah suatu algoritma yang menempatkan label kelas data sesuai dengan  $k$  divalidasi poin data yang sudah ditetapkan dengan perhitungan *K-Nearest Neighbor* (KNN). Dengan kata lain, proses MKNN pertama dengan melakukan perhitungan validitas data pada semua data latih. Selanjutnya, dilakukan perhitungan untuk mencari *weight voting* pada semua data uji menggunakan validitas data (Parvin, et al., 2010).

### 2.5.2 Proses Modified K-Nearest Neighbor (MKNN)

Secara garis besar terdapat dua proses utama dalam algoritma MKNN, yaitu (Parvin, et al., 2010) :

#### 2.5.2.1 Validitas Data Training

Validitas dari setiap data dihitung berdasarkan pada tetangganya. Validitas data dilakukan hanya sekali pada semua data latih. Setelah dilakukan validasi data selanjutnya data tersebut digunakan sebagai informasi tambahan (Parvin, et al., 2010). Untuk menghitung validitas data pada data training, tetangga terdekatnya perlu dipertimbangkan. Diantara tetangga terdekat dengan data, validitas digunakan untuk menghitung jumlah titik dengan label yang sama untuk data tersebut. Persamaan yang digunakan untuk menghitung validitas data training seperti pada persamaan (2.3) (Parvin, et al., 2010).

$$\text{validitas}(x) = \frac{1}{k} \sum_{i=1}^k S(\text{label}(x), (\text{label}(N_i(x)))) \quad (2.3)$$

Dimana:

$K$  = jumlah titik terdekat

Label (x) = kelas x

Label  $N_i(x)$  = label kelas titik terdekat x

Fungsi S digunakan untuk menghitung kesamaan antara titik x dan data ke-l dari tetangga terdekat. Fungsi S dijelaskan pada persamaan (2.4).

$$S(a, b) = \begin{cases} 1 & a = b \\ 0 & a \neq b \end{cases} \quad (2.4)$$

Dimana:

a = kelas a pada data *training*

b = kelas lain selain a pada data *training*

S akan bernilai 1, jika label kategori a sama dengan label kategori b. S akan bernilai 0, jika label kategori a tidak sama dengan label kategori b.

#### 2.5.2.2 Weight Voting KNN

*Weight voting* KNN adalah salah satu variasi metode KNN yang menggunakan k tetangga terdekat, terlepas dari kelas data, akan tetapi menggunakan *weight voting* dari masing-masing data pada data training. Masing-masing data diberikan *weight voting* yang biasanya sama dengan beberapa penurunan fungsi jarak dari data yang tidak diketahui. Persamaan *weight voting* dijelaskan pada persamaan sebagai berikut (Parvin, et al., 2010):

$$W_{(i)} = \frac{1}{d + \alpha} \quad (2.5)$$

Dimana d adalah jarak dan  $\alpha$  merupakan nilai *regulator smoothing*. Dalam penelitian ini nilai  $\alpha$  yang digunakan adalah 0,5. Kemudian, *weight voting* ini dijumlahkan setiap kelasnya dan kelas dengan jumlah terbesar yang akan dipilih menjadi sebuah keputusan. Selanjutnya, validitas dari tiap data pada data *training* dikalikan dengan *weight* berdasarkan pada jarak *Euclidean*. Dalam metode MKNN ini *weight voting* tiap tetangga dihitung menggunakan persamaan (2.6) (Parvin, et al., 2010).

$$W_{(i)} = \text{validitas}(i) \times \frac{1}{d_e + \alpha} \quad (2.6)$$

Dimana :

$W_{(i)}$  = perhitungan *weight voting*

*Validitas(i)* = Nilai *validitas*

$d_e$  = Jarak *Euclidean*

$\alpha$  = Nilai *regulator smoothing* (pemulusan)



Teknik *weight voting* ini mempunyai pengaruh yang lebih penting terhadap data yang mempunyai nilai validitas lebih tinggi dan paling dekat dengan data. Selain itu, dengan mengalikan validitas data dengan jarak dapat mengatasi kelemahan dari setiap data yang mempunyai jarak dengan *weight* yang memiliki banyak masalah dalam *outlier*. Jadi, algoritma MKNN direkomendasikan secara signifikan lebih bagus daripada metode KNN saja yang didasarkan hanya pada jarak saja (Parvin, et al., 2010).

## 2.6 Tanaman Bawang Merah

Bawang merah (*Allium cepa* L.) merupakan sayuran rempah yang cukup populer di Indonesia, memiliki nilai ekonomis tinggi, berfungsi sebagai penyedap rasa, dan dapat digunakan sebagai bahan obat tradisional. Prospek pengembangan bawang merah sangat baik, yang ditandai dengan meningkatnya konsumsi bawang merah seiring dengan bertambahnya jumlah penduduk. Rerata peningkatan konsumsi bawang merah perkapita per tahun dari tahun 2004 hingga 2008 mencapai 7,91 % (Departemen Pertanian, 2009).

Usahatani bawang merah merupakan usaha yang memiliki resiko tinggi, tidak sedikit tantangan dan kendala yang dihadapi dalam budidaya, seperti serangan OPT (Organisme Pengganggu Tumbuhan) yang dapat menggagalkan panen. Produktivitas tanaman yang rendah dan serangan hama/penyakit umumnya semakin meningkat pada pertanaman bawang merah diluar musim atau waktu *off-season*. Menurut Udiarto et al (2005) kehilangan hasil oleh serangan OPT pada tanaman bawang merah berkisar antara 20 sampai 100 persen. Dan OPT utama pada tanaman bawang merah yang menyerang sepanjang tahun, baik musim kemarau maupun musim hujan yaitu ulat bawang *Spodoptera exigua* (Moekasan et al, 2012).

## 2.7 Hama Penyakit Tanaman Bawang Merah

Dalam penelitian ini ada 5 jenis hama penyakit yang sering menyerang tanaman bawang merah yaitu sebagai berikut:

### 2.7.1 Hama Ulat Bawang (*Spodoptera Exigua*)

Daun bawang merah yang terserang kelihatan ada bercak putih panjang atau menjadi seperti membran dan layu. Warna ulat mula-mula hijau, kemudian menjadi coklat tua dengan garis-garis putih. Panjang ulat kurang lebih 2,5cm. Dengan gejala daging daun habis dan menjadi transparan. Ulat menyerang tanaman dengan memakan daun bagian dalam (Pracaya, 2009).

### 2.7.2 Hama Lalat Pengorok Daun (*Lirionzoa chinensis*)

Hama ini sejenis lalat yang meletakkan telur pada tanaman. Yang kemudian telur menetas menjadi larva yang menyerang daun bagian dalam dan membuat trowongan yang tampak bergaris-garis putih. Gejala serangan berat mulai pada umur 15 hari setelah tumbuh hingga menjelang tanam. Daun berwarna coklat seperti terbakar dan masuk ke dalam umbi



bawang. Hampir seluruh helain daun penuh kerokan atau terowongan(Pracaya, 2009).

### **2.7.3 Penyakit Moler atau Layu *Fusarium* (*Twisting Disease*)**

Bila penyakit ini terbawa pada bibit, gejala awal terlihat pada tanaman umur 5 – 10 hari setelah tanam. Jika penularan dari tanah, gejala tampak pada tanaman umur 3 minggu setelah tanam. Tanda adanya penyakit adalah tanaman menjadi cepat layu, akar tanaman busuk, tanaman terkulai seperti akan roboh, dan di dasar umbi lapis terlihat koloni jamur berwarna putih. Warna daun menjadi kuning dan bentuknya melengkung (moler).Tanaman kurus kekuningan dan busuk bagian pangkal/Sasaran serangan adalah bagian dasar dari umbi lapis. Daun bawang merah menguning dan terpelintir layu (moler) serta tanaman mudah tercabut karena pertumbuhan akar terganggu dan membusuk. Apabila umbi lapis dipotong membujur maka terlihat adanya pembusukan berawal dari dasar umbi meluas ke atas maupun ke samping(Baswarsiati, et al., 2017).

### **2.7.4 Penyakit Trotol atau Bercak Ungu (*Alternaria Porri*)**

Pada daun terdapat bercak berwarna kelabu ke ungu-unguan yang di dalamnya tampak garis melingkar seperti cincin. Bercak membesar membentuk cekungan. Cendawan juga menyerang umbi sehingga dapat menulari umbi lainnya. Kelembaban tinggi dan suhu rata-rata diatas 27°C membantu perkembangan penyakit. Jenis Sumenep lebih tahan. Penanaman bulan Juni-Juli dapat mengurangi serangan penyakit ini. Penyiraman tanaman setelah turun hujan dapat mengurangi serangan penyakit karena akan mencuci spora yang akan menempel pada daun bersama percikan air dari tanah(Baswarsiati, et al., 2017).

### **2.7.5 Penyakit Otomatis atau *Antraknose* (*Antracnose*)**

Di beberapa daerah penyakit ini disebut "otomatis" karena daun yang terserang melepuh, berwarna keputihan, kemudian daun patah secara serentak. Bila serangan ringan, pada hamparan terlihat spot-spot putih sporadis seperti gejala serangan hama tikus pada padi. Bila serangan berat seluruh hamparan menjadi putih dan panen gagal. Pergiliran tanaman dengan tanaman jenis bukan bawang-bawangan dan penanaman pada bulan Juni-Juli dapat mengurangi penyakit antraknosa(Baswarsiati, et al., 2017).

## **2.8 Akurasi Sistem**

Akurasi merupakan seberapa dekat suatu angka hasil pengukuran terhadap angka sebenarnya (true value atau reference value). Akurasi dapat diperoleh dari persentase kebenaran, yaitu perbandingan antara jumlah Diagnosis yang tepat dengan jumlah data keseluruhan. Akurasi dinyatakan dalam persamaan 2.7 (Sulistiyandari, 2012).

$$\text{akurasi} = \frac{\text{jumlah diagnosis benar}}{\text{jumlah data}} \times 100\%$$

(2.7)



## BAB 3 METODOLOGI

Bab ini menjelaskan tentang langkah-langkah yang akan dilakukan dalam penelitian. Penelitian dilakukan dengan melakukan beberapa tahapan yaitu studi literatur, pengumpulan data, *Analisis* kebutuhan, *Analisis* dan perancangan sistem, implementasi sistem, pengujian sistem serta *evaluasi* dan *analisis* hasil dari sistem yang akan dibuat. Berikut ini merupakan langkah-langkah pengerjaan yang diilustrasikan dalam diagram blok metode penelitian pada Gambar 3.1.



Gambar 3.1 Diagram Alir Metodologi Penelitian

### 3.1 Studi literatur

Tahapan ini dilakukan untuk mendapatkan dasar-dasar teori dan sumber acuan untuk pembuatan sistem Diagnosis penyakit pada tanaman bawang merah menggunakan metode *Modified K-Nearest Neighbor*, diantaranya sebagai berikut:

1. Teori Metode *Modified K-Nearest Neighbor*
2. Berbagai jenis hama penyakit pada tanaman bawang merah dan pengendaliannya

Literatur-literatur tersebut diperoleh dari buku, jurnal, penelitian sebelumnya dan situs-situs ilmiah. Cara ini dilakukan untuk mendapatkan dasar-dasar referensi yang kuat bagi peneliti untuk menyelesaikan penelitian.

### 3.2 Pengumpulan data

Lokasi penelitian ini di Balai Pengkajian Teknologi Pertanian (BPTP) Jawa Timur yang terletak di Karangploso, Malang, Jawa Timur. Penelitian ini menggunakan data latih yang digunakan sebagai acuan untuk pengembangan sistem. Data training yang digunakan terdiri dari 15 jenis gejala dari 5 jenis hama penyakit tanaman bawang merah. Data-data tersebut digunakan untuk menghitung tingkat keberhasilan dari sistem yang akan dibangun menggunakan metode *Modified K-Nearest Neighbor*.

Cara pengumpulan data untuk penelitian ini terdapat dua jenis data yaitu data primer dan data sekunder. Data primer adalah data yang didapatkan langsung dari responden penelitian. Sedangkan, data sekunder adalah data yang telah dikumpulkan oleh orang lain dan tidak dipersiapkan untuk kegiatan penelitian tetapi dapat digunakan untuk tujuan penelitian seperti melalui buku literatur atau yang lainnya. Metode pengumpulan data primer yang bersifat kuantitatif dapat menggunakan instrument kuisioner dan wawancara.

### 3.3 Analisis kebutuhan

Analisis kebutuhan merupakan sebuah fase yang berfungsi untuk merumuskan kebutuhan-kebutuhan yang diperlukan dalam mengembangkan suatu sistem. Analisis kebutuhan harus sesuai dengan lokasi penelitian, variabel penelitian dan mempersiapkan kebutuhan penelitian.

Secara keseluruhan, kebutuhan yang digunakan dalam pembuatan sistem diagnosa hama penyakit tanaman bawang merah ini meliputi:

1. Spesifikasi kebutuhan *hardware*, meliputi:
  - Laptop
2. Spesifikasi kebutuhan *software*, meliputi:
  - Windows 10 64bit sebagai sistem operasi
  - NetBeans
  - Android Studio
3. Data yang dibutuhkan, meliputi:
  - Data gejala penyakit tanaman bawang merah
  - Data nilai pembobotan gejala penyakit tanaman bawang merah

### 3.4 Analisis dan perancangan sistem

Analisis dan perancangan sistem membahas mengenai hal-hal yang terkait dengan apa saja yang diperlukan dalam pembuatan sistem. Hal-hal tersebut meliputi deskripsi sistem, perancangan program aplikasi, perhitungan manual, desain antarmuka dan perancangan pengujian. Hasil yang diharapkan pada perancangan sistem yaitu dapat membangun sebuah sistem berdasarkan hasil dan analisis yang ada. Secara umum, sistem yang dibangun adalah sebuah sistem untuk mendiagnosis hama penyakit tanaman bawang merah menggunakan metode *Modified K- Nearest Neighbor* (MKNN).

Pada tahap analisis, pertama menyiapkan data penyakit tanaman tomat yang telah berupa angka untuk digunakan dalam proses diagnosis penyakit. Model metode yang akan digunakan dalam sebuah diagram alir, agar permasalahan mudah untuk dipecahkan dan dilakukan perancangan sistem sesuai dengan analisis yang telah dilakukan.

Tahap perancangan sistem dilakukan untuk memecahkan permasalahan diagnosis hama penyakit tanaman bawang merah meliputi deskripsi umum sistem, perancangan model MKNN, perancangan proses MKNN dengan diagram alir, contoh perhitungan manual, perancangan antar muka dan perancangan pengujian sistem yang disajikan dalam bentuk tabel pengujian.

### 3.5 Implementasi

Implementasi perangkat lunak yang menerapkan algoritma *Modifed K-Nearest Neighbor* dilakukan dengan mengacu pada perancangan sistem yang dibuat. Implementasi sistem tersebut meliputi :

1. Implementasi *interface*, menggunakan *software* Android Studio.
2. Implementasi algoritma, melakukan perhitungan dengan metode *Modified K-Nearest Neighbor* menggunakan bahasa pemrograman *java* di NetBeans ke dalam *software* Android Studio.
3. Implementasi ini akan menghasilkan diagnosis hama penyakit yang menyerang tanaman bawang merah melalui masukan yang berupa gejala penyakit.

### 3.6 Pengujian Sistem

Pengujian sistem ini dilakukan agar dapat memastikan bahwa aplikasi yang telah dibangun dapat bekerja sesuai dengan tujuan yang diharapkan. Pengujian yang dilakukan ada 2 macam, yaitu:

1. Pengujian pengaruh nilai k  
Pengujian ini dilakukan untuk mengetahui apakah nilai k berpengaruh terhadap akurasi yang dihasilkan sistem.
2. Pengujian akurasi sistem  
Pengujian akurasi sistem ini dilakukan untuk mengetahui apakah

sistem cukup baik untuk digunakan, dan juga untuk mengetahui seberapa besar akurasi metode MKNN ini dalam penerapan diagnosis hama penyakit tanaman bawang merah.

### 3.7 Evaluasi

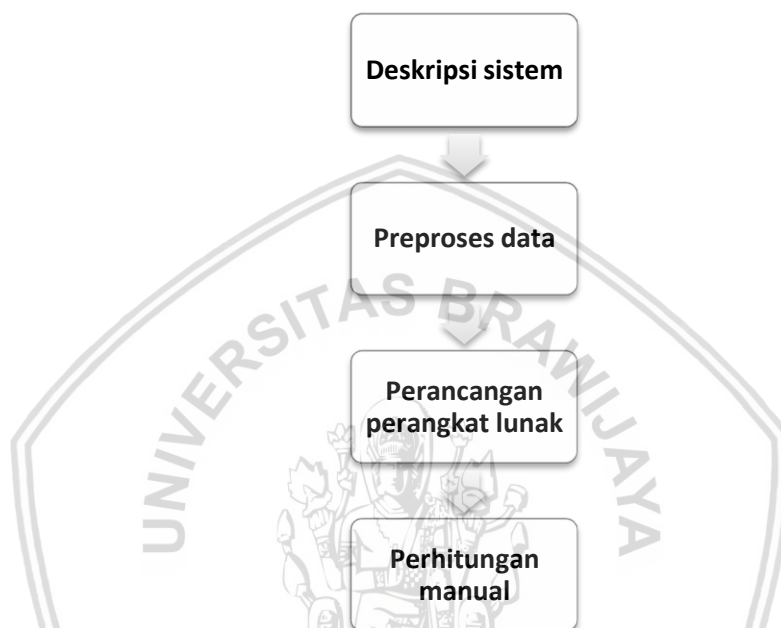
Tahap evaluasi dilakukan setelah proses pengujian sistem, evaluasi dilakukan untuk mempertimbangkan tingkat keberhasilan aplikasi dari implementasi metode dalam menyelesaikan permasalahan Diagnosis hama penyakit tanaman bawang merah dan apakah kinerja metode MKNN lebih optimal sesuai dengan tujuan penelitian ini. Tahap akhir dari seluruh rangkaian proses yaitu proses penarikan kesimpulan terhadap hasil yang diperoleh dari proses pengujian dan analisis yang telah dievaluasi. Selanjutnya kesimpulan dan saran dapat digunakan oleh pembaca sebagai acuan untuk perbaikan serta pengembangan aplikasi ke depannya.





## BAB 4 PERANCANGAN

Bab ini membahas tentang perancangan pada sistem untuk Diagnosis Hama Penyakit Tanaman Bawang Merah Menggunakan Metode *Modified K-Nearest Neighbor*. proses perancangan sistem terdiri dari deskripsi sistem, *preprocessing* data, perancangan perangkat lunak, perhitungan manual, desain antarmuka dan perancangan pengujian. Diagram alir proses perancangan sistem dapat dilihat pada Gambar 4.1.



Gambar 4.1 Diagram Alir Tahap Perancangan Sistem

### 4.1 Deskripsi sistem

Sistem yang akan dibuat dalam skripsi ini bertujuan mengimplementasikan metode *Modified K-Nearest Neighbor* dan mengetahui kinerja metode tersebut dalam permasalahan diagnosis hama penyakit tanaman bawang merah. Hama penyakit yang dapat didiagnosis dalam sistem ada 5 macam yaitu hama ulat bawang, lalat pengorok daun, penyakit moler, penyakit trotol, dan *antraknose*. Masukan yang digunakan dalam sistem ini berupa 15 pertanyaan gejala hama penyakit yang menyerang tanaman bawang merah dan kemudian jawaban dari masukan tersebut akan diklasifikasikan menggunakan metode MKNN.

Pada sistem yang akan dibuat ini terdapat dua tahap utama yaitu pelatihan data dan pengujian data. Proses pelatihan data dilakukan untuk mendapatkan nilai validitas dari data latih yang kemudian digunakan untuk proses pengujian. Sedangkan, pada proses pengujian akan didapatkan hasil diagnosis hama penyakit menurut sistem dan kemudian hasil tersebut dibandingkan dengan hasil diagnosis pada data asli dari pakar untuk mendapatkan nilai akurasi.

## 4.2 Preproses data

Preproses data merupakan suatu tahapan yang dilakukan untuk menyeleksi data dari semua alternatif yang telah terkumpul dalam tahap pengumpulan data. Alternatif data yang telah diseleksi kemudian dipisahkan sesuai dengan kriteria- kriteria yang telah ditentukan. Kriteria atau parameter yang digunakan pada data yang akan digunakan terdiri dari 15 gejala hama penyakit tanaman bawang merah. Setiap gejala penyakit ini diberikan nilai pembobotan dari pakar untuk mempermudah proses perhitungan pada sistem. Nilai pembobotan untuk setiap gejala berbeda-beda, setiap gejala memiliki tiga opsi nilai yaitu tidak ada, sedikit dan banyak. Nilai pembobotan dari 15 gejala penyakit tanaman tomat ditunjukkan pada Tabel 4.1.

**Tabel 4.1 Nilai Pembobotan 15 Gejala Hama Penyakit Tanaman Bawang Merah Dari Pakar**

No	Kode	Gejala	Opsi	Nilai
1	G1	Terdapat Ulat <i>Spodoptera exigua</i> berukuran sekitar 25 mm, berwarna hijau atau coklat dengan garis tengah berwarna kuning.	tidak ada	0
			sedikit	45
			banyak	90
2	G2	Daging daun habis dan menjadi transparan.	tidak ada	0
			sedikit	10
			banyak	20
3	G3	Terdapat ulat yang menyerang tanaman dengan memakan daun bagian dalam.	tidak ada	0
			sedikit	10
			banyak	20
4	G4	Terdapat Lalat Pengorok, gejala serangan mulai pada umur 15 hari setelah tumbuh hingga menjelang panen.	tidak ada	0
			sedikit	10
			banyak	20
5	G5	Daun berwarna coklat seperti terbakar dan masuk ke dalam umbi bawang.	tidak ada	0
			sedikit	10
			banyak	20
6	G6	Hampir seluruh helaian daun penuh kerokan.	tidak ada	0
			sedikit	45
			banyak	90
7	G7	Daun bawang merah menguning dan terpelintir layu (moler).	tidak ada	0
			sedikit	45

			banyak	90
8	G8	Tanaman mudah tercabut karena pertumbuhan akar terganggu dan membusuk.	tidak ada	0
			sedikit	10
			banyak	20
9	G9	Apabila umbi lapis dipotong membujur akan terlihat adanya pembusukan berawal dari dasar umbi meluas ke atas dan ke samping.	tidak ada	0
			sedikit	10
			banyak	20
10	G10	Tanaman kurus kekuningan dan busuk bagian pangkal.	tidak ada	0
			sedikit	10
			banyak	20
11	G11	Ujung daun mengering dan daun patah.	tidak ada	0
			sedikit	10
			banyak	20
12	G12	Daun timbul bercak berukuran kecil, berwarna putih dengan pusat berwarna ungu.	tidak ada	0
			sedikit	45
			banyak	90
13	G13	Terbentuk Lekukan ke dalam, berlubang dan patah.	tidak ada	0
			sedikit	45
			banyak	90
14	G14	Terdapat bercak berwarna putih pada daun.	tidak ada	0
			sedikit	10
			banyak	20
15	G15	Tanaman akan mati dengan cepat dan mendadak.	tidak ada	0
			sedikit	10
			banyak	20

Keterangan :

- 0 : tidak ada gejala  
 10 : gejala sedikit tapi bukan gejala utama  
 20 : gejala banyak tapi bukan gejala utama atau  
 45 : gejala sedikit dan merupakan gejala utama  
 90 : gejala banyak dan merupakan gejala utama

15 hama penyakit bawang merah tersebut mewakili dari 5 hama penyakit utama yang akan didiagnosis pada sistem menggunakan metode MKNN. Pasangan gejala hama penyakit dan hama penyakit yang menyerang tanaman bawang merah dapat dilihat pada Tabel 4.2.

**Tabel 4.2 Pasangan Gejala dan Penyakit yang Menyerang Bawang Merah**

G1	G2	G3	G4	G5	G6	G7	G8	G9	G10	G11	G12	G13	G14	G15	Diagnosis
X	X	X													Ulat Bawang
			X	X	X										Lalat Pengorok
						X	X	X	X						Moler
										X	X				Trorol
												X	X	X	Antraknose



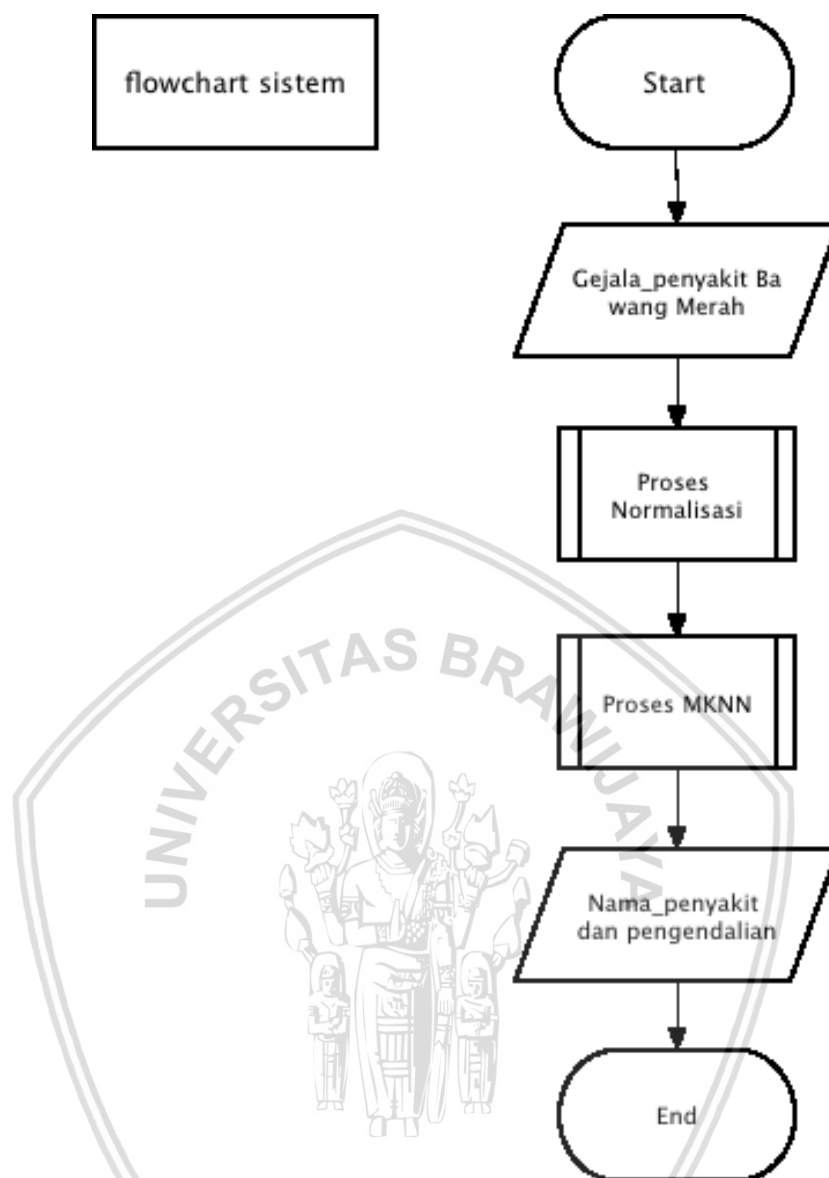
Keterangan : Gejala utama penyakit

### 4.3 Perancangan perangkat lunak

Perancangan perangkat lunak ini menjelaskan mengenai hubungan antar komponen-komponen sehingga mampu membentuk sebuah fungsi yang mampu memberikan pelayanan terhadap kebutuhan pengguna. Perancangan perangkat lunak pada aplikasi untuk diagnosis hama penyakit tanaman bawang merah menggunakan metode MKNN terdiri dari perancangan database dan perancangan proses.

#### 4.3.1 Perancangan proses

Proses diagnosis hama penyakit tanaman bawang merah dilakukan dengan memasukkan data gejala-gejala hama penyakit tanaman bawang merah yang sudah tersimpan pada sistem. Proses memasukkan data gejala-gejala hama penyakit tersebut dilakukan dengan cara menjawab pertanyaan yang telah disediakan pada interface menu awal. Setelah selesai memasukkan gejala dan menekan tombol diagnosis, maka dapat dilakukan proses diagnosis perhitungan dengan menggunakan *metode Modified K-Nearest Neighbor* dan sistem akan menunjukkan hasil diagnosis berupa nama penyakit beserta pengendaliannya. Sedangkan pada menu data latih, pengguna memasukkan jumlah data latih dan nilai  $k$  yang akan digunakan untuk perhitungan. Setelah itu, pengguna dapat melihat hasil perhitungan dari nilai validitas, nilai *weight voting* dan hasil klasifikasi. Pada halaman pengujian pengguna memasukkan data uji yang akan diklasifikasikan, kemudian data tersebut akan dihitung dan hasil diagnosis dari sistem akan dicocokkan dengan hasil diagnosis asli untuk dicari nilai akurasi. Diagram alir sistem dapat dilihat pada Gambar 4.2.

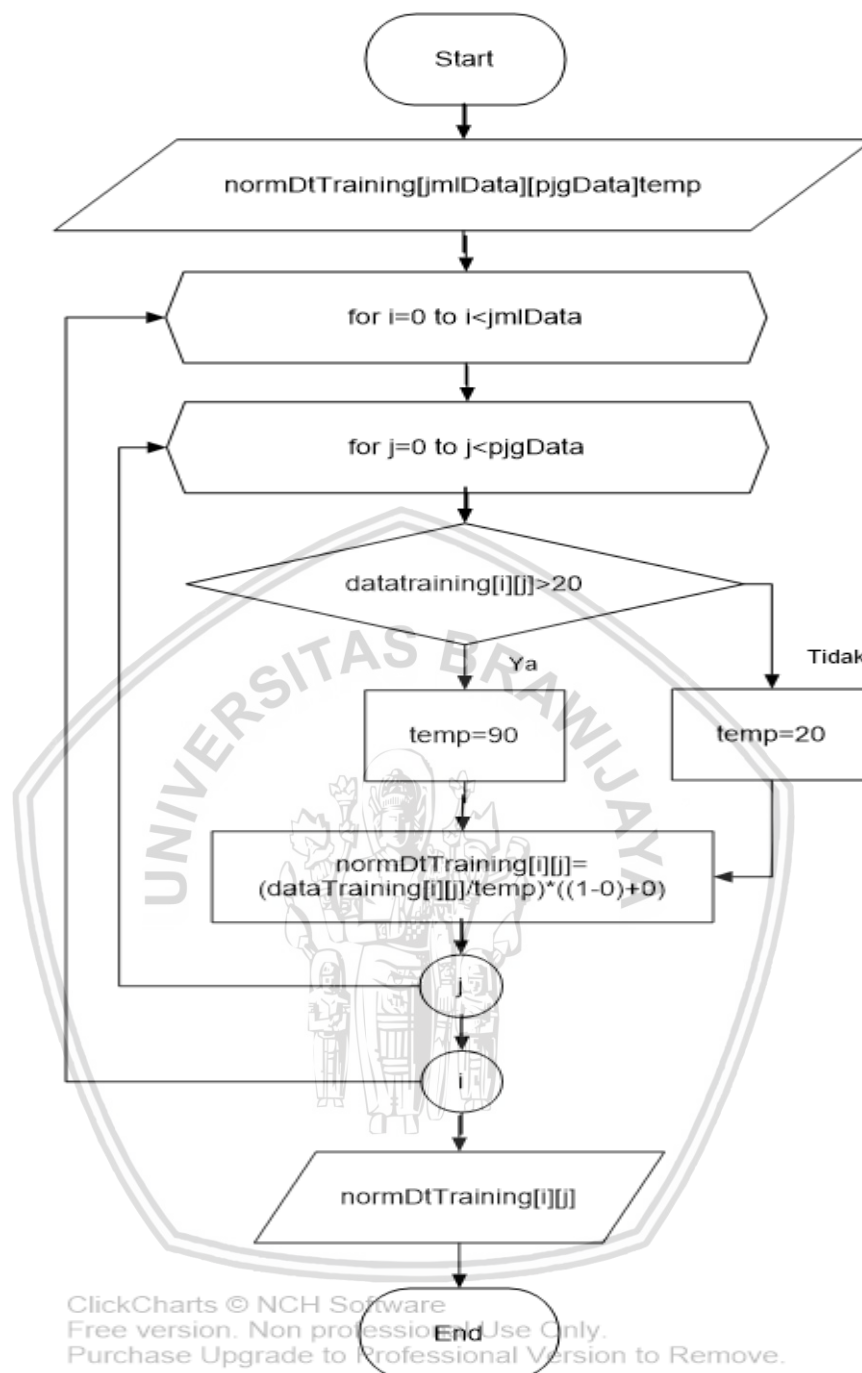


Gambar 4.2 Diagram Alir Sistem

#### 4.3.1.1 Proses perhitungan normalisasi data

Proses normalisasi data ini akan menormalisasikan data latih dan data uji yang akan digunakan dalam proses perhitungan. Hal ini bertujuan agar data berada pada range  $[0,1]$  sehingga sebaran data tidak terlalu jauh. Persamaan 2.2 digunakan pada proses normalisasi data latih dan data uji gejala penyakit tanaman tomat. Proses normalisasi data dapat dilihat pada Gambar 4.3.

Pada diagram alir Gambar 4.3 proses normalisasi dimulai dengan menginputkan data gejala penyakit tanaman bawang merah. Kemudian data tersebut dihitung menggunakan persamaan 2.2 dan hasilnya normalisasi ini digunakan untuk proses perhitungan selanjutnya.

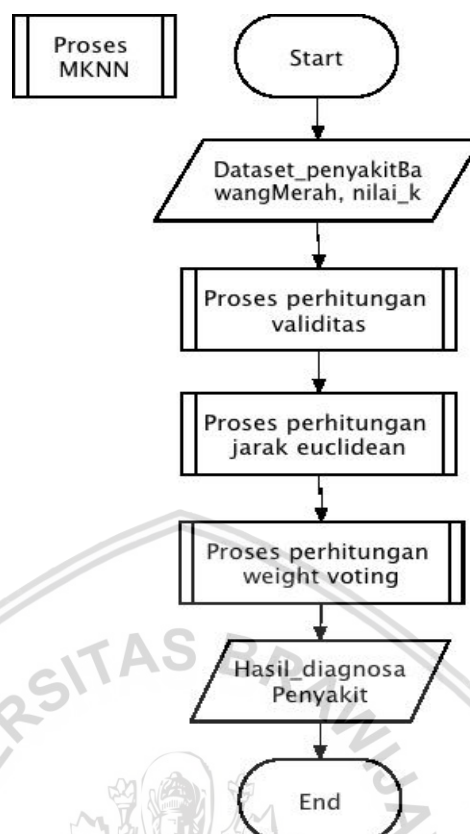


**Gambar 4.3 Diagram Alir Proses Perhitungan Normalisasi**

#### 4.3.1.2 Proses Algoritma Modified K-Nearest Neighbor

Proses algoritma Modified K-Nearest Neighbor terdiri dari 3 proses utama, yaitu proses perhitungan validitas, proses perhitungan jarak Euclidean, dan perhitungan weight voting. Proses algoritma MKNN dapat dilihat pada Gambar 4.4.





**Gambar 4.4 Diagram Alir Proses Algoritma MKNN**

Pada algoritma MKNN tahapan awal dimulai dengan memasukkan jumlah data latih yang akan digunakan dan nilai  $k$ . selanjutnya, dilakukan proses validitas terhadap data latih. Setelah melakukan validitas data latih dilakukan proses perhitungan jarak Euclidean terlebih dahulu. Proses selanjutnya yaitu melakukan perhitungan jarak antar data uji dan data latih. Setelah proses tersebut selesai, maka dilakukan proses perhitungan *weight voting*. Tahapan terakhir didapatkan dengan menghitung nilai *weight voting* terbesar sesuai nilai  $k$  yang dimasukkan, maka akan dihasilkan keluaran hasil diagnosis penyakit tanaman bawang merah.

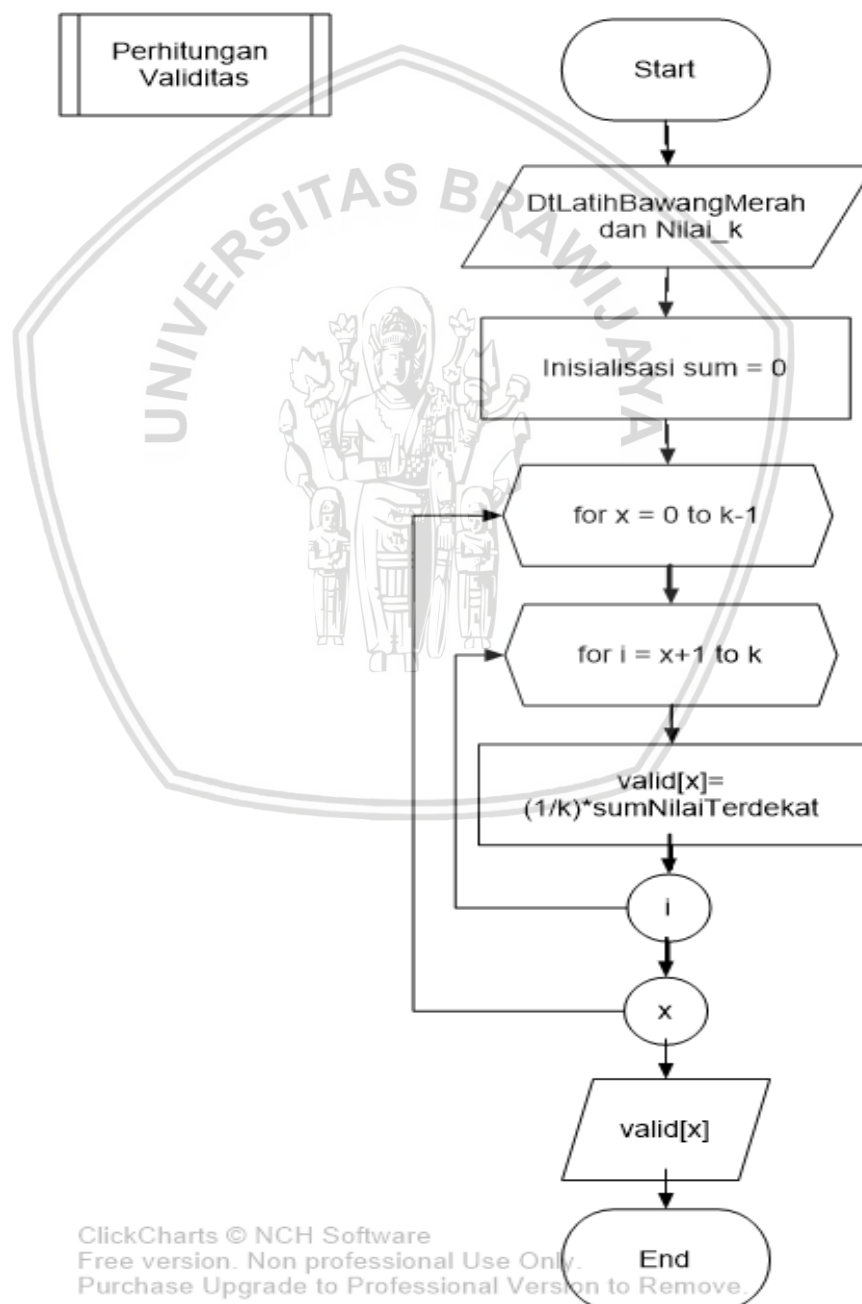
#### 4.3.1.3 Proses perhitungan validitas

Proses perhitungan validitas ini merupakan suatu proses melakukan validitas untuk tiap-tiap data latih. Validitas data latih ini dilakukan untuk memperoleh informasi lebih mengenai data yang akan digunakan untuk proses *weight voting*. Proses perhitungan validitas dapat dilihat pada Gambar 4.5.

Pada diagram alir perhitungan validitas terdiri dari beberapa tahapan. Tahapan awal yaitu memasukkan jumlah data latih penyakit tanaman bawang merah dan nilai  $k$ . Kemudian, dilakukan perhitungan validitas dengan membandingkan kelas kategori penyakit tanaman bawang merah pada data latihnya sesuai dengan nilai  $k$  yang dimasukan. Perhitungan ini

dilakukan berdasarkan dengan persamaan 2.3. Proses yang terdapat pada perhitungan manual validitas data latih ini dapat dilihat pada Tabel 4.19.

Proses validitas yang dilakukan adalah dengan melakukan input data latih penyakit tanaman bawang merah kemudian membandingkan kelas-kelas pada data latihnya sesuai dengan ketentuan sebelumnya. Berdasarkan persamaan 2.4, pada proses validitas terdapat ketentuan, jika kelasnya sama maka nilainya adalah 1 dan jika kelasnya tidak sama maka nilainya 0 serta dilakukan perbandingan data sebanyak k. Kemudian  $valid[x]$  akan dijumlah dan dibagi sebanyak k data yang telah diinputkan. Maka akan didapatkan output data nilai validitas tiap data uji pada proses klasifikasi algoritma MKNN.



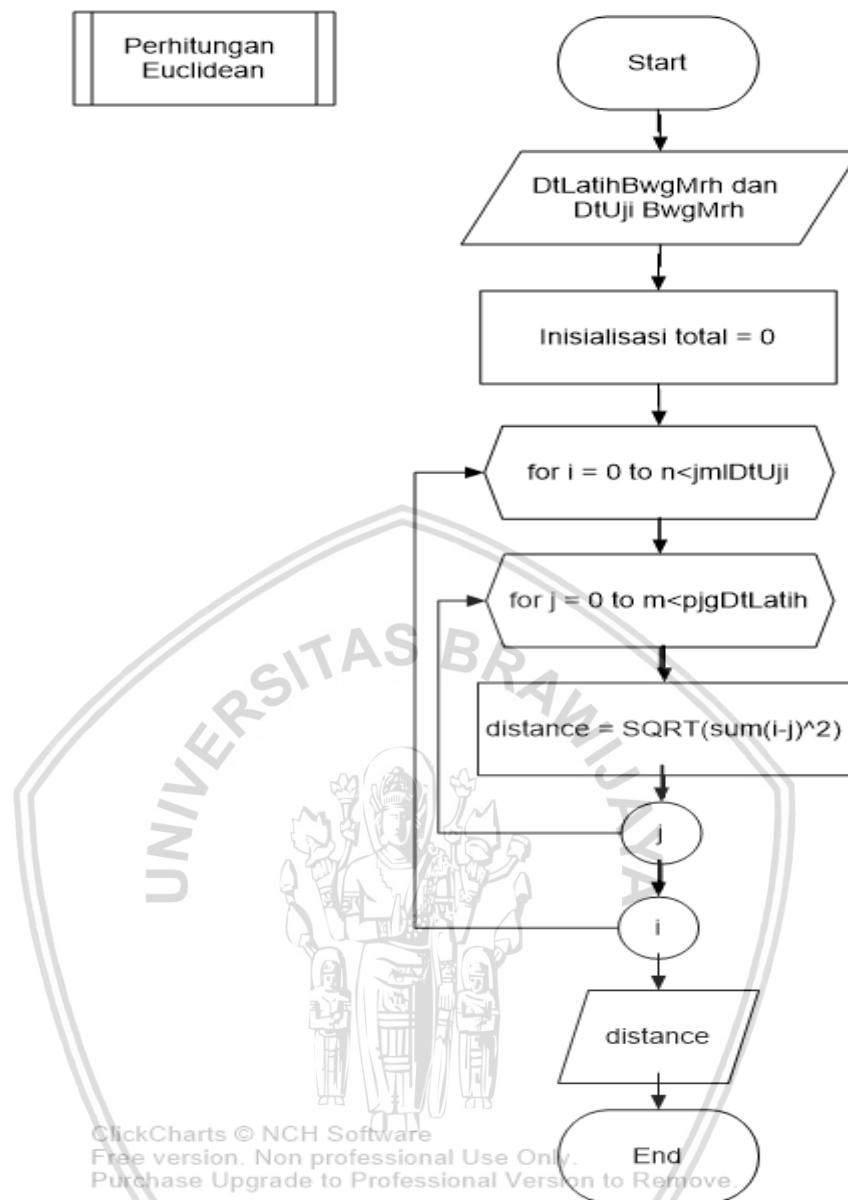
**Gambar 4.5 Diagram Alir Proses Perhitungan Validitas**

#### **4.3.1.4 Proses Perhitungan Jarak Euclidean**

Proses perhitungan jarak *Euclidean* ini dilakukan untuk menghitung jarak antar data latih serta jarak antar data latih dan data uji pada penyakit tanaman bawang merah. Perhitungan jarak tersebut berdasarkan atribut penyakit tanaman bawang merah. Diagram alir proses perhitungan jarak *Euclidean* dapat dilihat pada Gambar 4.6.

Pada diagram alir proses perhitungan jarak *euclidean* tahapan pertama yang dilakukan adalah melakukan masukan data gejala penyakit tanaman bawang merah. Masukan data tersebut akan dilakukan perhitungan *euclidean* sesuai dengan persamaan 2.1 sebanyak data inputan. Setelah itu, akan didapatkan keluaran data nilai *euclidean* setiap data uji pada proses MKNN.

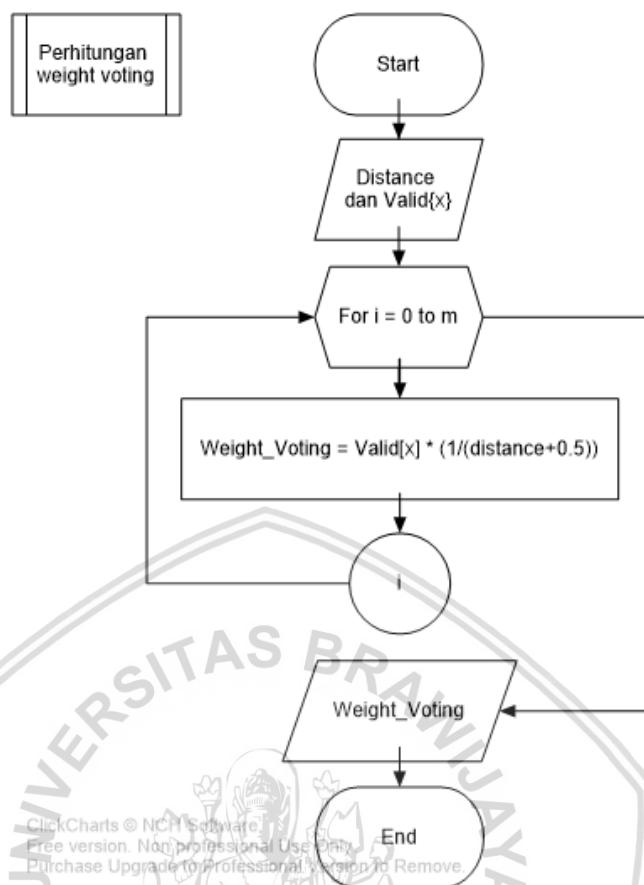




Gambar 4.6 Diagram Alir Proses Perhitungan Jarak *Euclidean*

#### 4.3.1.5 Perancangan Proses Perhitungan *Weight voting*

Pada proses perhitungan *weight voting* terlebih dahulu memasukkan nilai hasil *validitas* masing-masing data latih yang diperoleh dari perhitungan jarak *euclidean*. Selanjutnya, akan dilakukan proses perhitungan *weight voting* dengan persamaan 2.6 sebanyak data uji. Dari proses ini akan dihasilkan keluaran nilai *weight voting* dari tiap data uji pada proses klasifikasi algoritma MKNN. Proses perhitungan *weight voting* dapat dilihat pada Gambar 4.7.



Gambar 4.7 Proses Perhitungan *Weight voting*

#### 4.4 Perhitungan manual

Perhitungan manual berfungsi untuk memberikan gambaran umum perancangan sistem yang dibangun. Dari metode *Modified K-Nearest Neighbor* (MKNN) langkah-langkah dalam perhitungannya yaitu:

1. Menentukan nilai k atau tetangganya
2. Melakukan normalisasi data
3. Menghitung validitas data latih
4. Menghitung jaran *enclidean*
5. Menghitung *weight voting* (pembobotan)
6. Menentukan kelas dari data uji

Pada perhitungan manual ini, hanya mengambil 15 dataset acak dari 74 dataset dari pakar dimana dibagi menjadi dua bagian yaitu 10 data latih dan 5 data uji, untuk perwakilan setiap penyakitnya ada 2. Dataset yang digunakan dapat dilihat pada Tabel 4.3 berikut.

Tabel 4.3 Dataset Perhitungan Manual

No	G1	G2	G3	G4	G5	G6	G7	G8	G9	G10	G11	G12	G13	G14	G15	kelas
----	----	----	----	----	----	----	----	----	----	-----	-----	-----	-----	-----	-----	-------

1	90	10	20	0	20	0	45	0	20	0	10	0	0	0	20	1
2	45	20	20	0	20	0	0	10	0	10	0	45	0	0	10	1
3	0	10	0	20	10	90	0	0	10	0	10	0	0	20	0	2
4	0	0	10	20	20	45	0	0	20	0	10	45	0	0	20	2
5	0	0	0	10	0	0	90	10	20	20	0	0	0	10	0	3
6	0	20	0	0	0	0	45	20	20	20	0	45	0	0	0	3
7	0	20	0	10	0	0	0	0	10	0	10	90	0	0	0	4
8	0	0	0	20	0	45	0	0	0	0	20	45	0	10	0	4
9	0	0	10	0	0	0	0	20	0	10	0	45	90	10	20	5
10	0	0	0	10	20	0	0	0	0	10	10	0	45	20	20	5
11	90	20	20	0	0	0	0	10	0	0	0	0	10	0	0	1
12	0	0	0	20	20	90	0	0	0	0	0	0	0	10	0	2
13	0	10	0	0	0	0	90	20	20	20	0	0	0	0	0	3
14	0	0	0	10	0	0	0	10	0	0	20	90	0	0	0	4
15	0	0	10	0	0	0	0	20	0	0	0	90	90	20	20	5

Dataset yang digunakan terdiri dari 15 gejala penyakit dan diagnosis dari tiap data. Pembagian data latih dan data uji dapat dilihat pada Tabel 4.4 dan Tabel 4.5 berikut.

**Tabel 4.4 Data Latih**

No	G1	G2	G3	G4	G5	G6	G7	G8	G9	G10	G11	G12	G13	G14	G15	kelas
1	90	10	20	0	20	0	45	0	20	0	10	0	0	0	20	1
2	45	20	20	0	20	0	0	10	0	10	0	45	0	0	10	1
3	0	10	0	20	10	90	0	0	10	0	10	0	0	20	0	2
4	0	0	10	20	20	45	0	0	20	0	10	45	0	0	20	2
5	0	0	0	10	0	0	90	10	20	20	0	0	0	10	0	3
6	0	20	0	0	0	0	45	20	20	20	0	45	0	0	0	3
7	0	20	0	10	0	0	0	0	10	0	10	90	0	0	0	4
8	0	0	0	20	0	45	0	0	0	0	20	45	0	10	0	4
9	0	0	10	0	0	0	0	20	0	10	0	45	90	10	20	5
10	0	0	0	10	20	0	0	0	0	10	10	0	45	20	20	5

**Tabel 4.5 Data Uji**

No	G1	G2	G3	G4	G5	G6	G7	G8	G9	G10	G11	G12	G13	G14	G15	kelas
1	90	20	20	0	0	0	0	10	0	0	0	0	10	0	0	1
2	0	0	0	20	20	90	0	0	0	0	0	0	0	10	0	2



3	0	10	0	0	0	0	90	20	20	20	0	0	0	0	0	3
4	0	0	0	10	0	0	0	10	0	0	20	90	0	0	0	4
5	0	0	10	0	0	0	0	20	0	0	0	90	90	20	20	5

**Langkah 1.** Menentukan nilai k atau tetangganya.

Pada perhitungan manual ini ditentukan nilai dari k yaitu 3

**Langkah 2.** Melakukan proses normalisasi data.

Pada langkah ini dilakukan proses normalisasi data untuk menyamakan sebaran data latih maupun data uji. Perhitungan normalisasi menggunakan persamaan 2.2 yaitu perhitungan min-max (Han & Kamber, 2006). Data latih sebelum dinormalisasi dapat dilihat pada Tabel 4.4 dan data uji sebelum dinormalisasi dapat dilihat pada Tabel 4.5. Sedangkan data latih hasil normalisasi dapat dilihat pada Tabel 4.6 dan data uji hasil normalisasi dapat dilihat pada Tabel 4.7.

**Tabel 4.6 Data Latih Hasil Normalisasi**

No	G1	G2	G3	G4	G5	G6	G7	G8	G9	G10	G11	G12	G13	G14	G15	kelas
1	1	0,5	1	0	1	0	0,5	0	1	0	0,5	0	0	0	1	1
2	0,5	1	1	0	1	0	0	0,5	0	0,5	0	0,5	0	0	0,5	1
3	0	0,5	0	1	0,5	1	0	0	0,5	0	0,5	0	0	1	0	2
4	0	0	0,5	1	1	0,5	0	0	1	0	0,5	0,5	0	0	1	2
5	0	0	0	0,5	0	0	1	0,5	1	1	0	0	0	0,5	0	3
6	0	1	0	0	0	0	0,5	1	1	1	0	0,5	0	0	0	3
7	0	1	0	0,5	0	0	0	0	0,5	0	0,5	1	0	0	0	4
8	0	0	0	1	0	0,5	0	0	0	0	1	0,5	0	0,5	0	4
9	0	0	0,5	0	0	0	0	1	0	0,5	0	0,5	1	0,5	1	5
10	0	0	0	0,5	1	0	0	0	0	0,5	0,5	0	0,5	1	1	5

**Tabel 4.7 Data Uji Hasil Normalisasi**

No	G1	G2	G3	G4	G5	G6	G7	G8	G9	G10	G11	G12	G13	G14	G15	kelas
1	1	1	1	0	0	0	0	0,5	0	0	0	0	0,5	0	0	1
2	0	0	0	1	1	1	0	0	0	0	0	0	0	0,5	0	2
3	0	0,5	0	0	0	0	1	1	1	1	0	0	0	0	0	3
4	0	0	0	0,5	0	0	0	0,5	0	0	1	1	0	0	0	4

5	0	0	0.5	0	0	0	0	1	0	0	0	1	1	1	1	5
---	---	---	-----	---	---	---	---	---	---	---	---	---	---	---	---	---

### Langkah 3. Menghitung validitas data latih.

Langkah ketiga yaitu menghitung nilai validitas data latih dengan menggunakan persamaan 2.3. Sebelum menghitung nilai validitas, terlebih dahulu mencari jarak data antar data latih menggunakan perhitungan *Euclidean* dengan memasukkan data pada persamaan 2.1.

$$d_{(1,2)} = \sqrt{\sum (x_{1i} - x_{2i})^2}$$

$$\begin{aligned}
 &= \sqrt{(1-0.5)^2 + (0.5-1)^2 + (1-1)^2 + (0-0)^2 + (1-1)^2 + (0-0)^2} \\
 &\quad + (0.5-0)^2 + (0-0.5)^2 + (1-0)^2 + (0-0.5)^2 + (0.5-0)^2 + (0-0.5)^2 \\
 &\quad + (0-0)^2 + (0-0)^2 + (1-0.5)^2 \\
 &= 1.73
 \end{aligned}$$

Melakukan perhitungan yang sama untuk semua data latih. Hasil perhitungan *Euclidean* data latih ini dapat dilihat pada Tabel 4.8.

**Tabel 4.8 Perhitungan *Euclidean* Data Latih**

d1	d2	d3	d4	d5	d6	d7	d8	d9	d10	kelas
0	1.73	2.50	1.80	2.55	2.60	2.45	2.74	2.60	2.29	1
1.73	0	2.50	2.18	2.55	2.06	2.00	2.55	2.06	2.18	1
2.60	2.50	0	1.87	2.18	2.55	1.94	1.32	2.65	1.87	2
1.80	2.18	1.87	0	2.40	2.65	2.06	1.94	2.45	1.87	2
2.55	2.55	2.18	2.40	0	1.50	2.24	2.24	2.29	2.29	3
2.60	2.06	2.55	2.65	1.50	0	1.80	2.60	2.24	2.74	3
2.45	2.00	1.94	2.06	2.24	1.80	0	1.58	2.40	2.40	4
2.74	2.55	1.32	1.94	2.24	2.60	1.58	0	2.40	1.94	4
2.60	2.06	2.60	2.45	2.29	2.24	2.40	2.40	0	1.87	5
2.29	2.18	2.60	1.87	2.29	2.74	2.40	1.94	1.87	0	5

Keterangan :



Jarak *Euclidean* yang memiliki ketetanggaan terdekat dengan data

Setelah ditentukan nilai k nya maka dihitung nilai validitas dari data latih yang memiliki jarak terdekat dengan persamaan 2.3.

$$\text{Validitas}(x) = 1/k \sum S(\text{label}(x=1), (\text{label}(N_i(x=2))))$$

$$=1/3 (1+0+0)$$

$$=0.33$$

Melakukan perhitungan yang sama untuk semua data latih. Hasil perhitungan validitas ini dapat dilihat pada Tabel 4.9 berikut.

**Tabel 4.9 Perhitungan Validitas**

No	k=1	k=2	k=3	Sum S(a,b)	Validitas
1	1	0	0	1	0.33
2	1	0	0	1	0.33
3	1	0	0	1	0.33
4	1	0	0	1	0.33
5	1	0	0	1	0.33
6	1	0	0	1	0.33
7	1	0	0	1	0.33
8	1	0	0	1	0.33
9	1	0	0	1	0.33
10	1	0	0	1	0.33

**Langkah 4.** Menghitung jarak *euclidean* data uji

Pada langkah selanjutnya adalah mencari nilai *euclidean* dengan memasukkan data ujipada persamaan 2.1, dimana contoh perhitungan data uji pertama sebagai berikut

$$\begin{aligned}
 d_{(1,1)} &= \sqrt{\sum (x_{1i} - x_{2i})^2} \\
 &= \sqrt{(1-1)^2 + (1-0.5)^2 + (1-1)^2 + (0-0)^2 + (0-1)^2 + (0-0)^2} \\
 &\quad + (0-0.5)^2 + (0.5-0)^2 + (0-1)^2 + (0-0)^2 + (0-0.5)^2 + (0-0)^2 \\
 &\quad + (0.5-0)^2 + (0-0)^2 + (0-1)^2 \\
 &= 2.06
 \end{aligned}$$

Hasil perhitungan *Euclidean* ini dapat dilihat pada tabel 4.10 berikut.

**Tabel 4.10 Tabel Perhitungan *Euclidean* Data Uji 1**

Data	SUM Euclidean	Euclidean
d(1,1)	4.25	2.06155281
d(1,2)	2.25	1.5

d(1,3)	6.5	2.55
d(1,4)	7.5	2.74
d(1,5)	6.75	2.59807621
d(1,6)	5	2.23606798
d(1,7)	4.25	2.06
d(1,8)	6.25	2.50
d(1,9)	4.5	2.12
d(1,10)	7	2.65

#### Langkah 5. Menghitung *weight voting*

Dalam penelitian ini nilai  $\alpha$  yang digunakan adalah 0,5. Kemudian, *weight voting* ini dijumlahkan setiap kelasnya dan kelas dengan jumlah terbesar yang akan dipilih menjadi sebuah keputusan. Selanjutnya, validitas dari tiap data pada data training dikalikan dengan weight berdasarkan pada jarak Euclidean. Dalam metode MKNN ini *weight voting* tiap tetangga dihitung menggunakan persamaan (2.6).

$$W(i) = \text{validitas}(i) \times 1/(d + \alpha)$$

$$= 0.33 \times 1/(2.06 + 0.5)$$

$$= 0.13013$$

Melakukan perhitungan yang sama untuk semua data. Hasil perhitungan validitas ini dapat dilihat pada Tabel 4.11 berikut.

**Tabel 4.11 Tabel Nilai *Weight Voting* Data Uji 1 dengan 3 Nilai Terbesar**

Data	Weight Voting	Diagnosis
d(1,1)	0.13013	1
d(1,2)	0.16667	1
d(1,3)	0.10931	2
d(1,4)	0.10292	2

d(1,5)	0.10759	3
d(1,6)	0.12183	3
d(1,7)	0.13013	4
d(1,8)	0.11111	4
d(1,9)	0.12716	5
d(1,10)	0.10596	5

#### Langkah 6. Menentukan kelas dari data uji

Setelah didapatkan nilai *weight voting* dari semua data latih, maka dilakukan pencarian nilai *weight voting* yang terbesar sebanyak nilai k yang telah ditentukan. Hasil pencarian nilai *weight voting* terbesar dapat dilihat pada tabel 4.11. Dari Tabel 4.11 maka didapatkan 3 *weight voting* terbesar yaitu 0.13013 dengan kelas 1, 0.16667 dengan kelas 1, dan 0.13013 dengan kelas 4. Setelah diambil nilai *weight voting* terbesar sebanyak 3, maka selanjutnya dilakukan penjumlahan nilai *weight voting* antara kelas yang sama. Dari data uji 1, maka kelas kategori 1 jumlah *weight voting*nya adalah  $0.13013 + 0.16667 = 0.2968$  sehingga data uji 1 kelas kategorinya adalah kelas 1 karena nilai *weight voting* nya lebih besar dari kategori kelas lain.

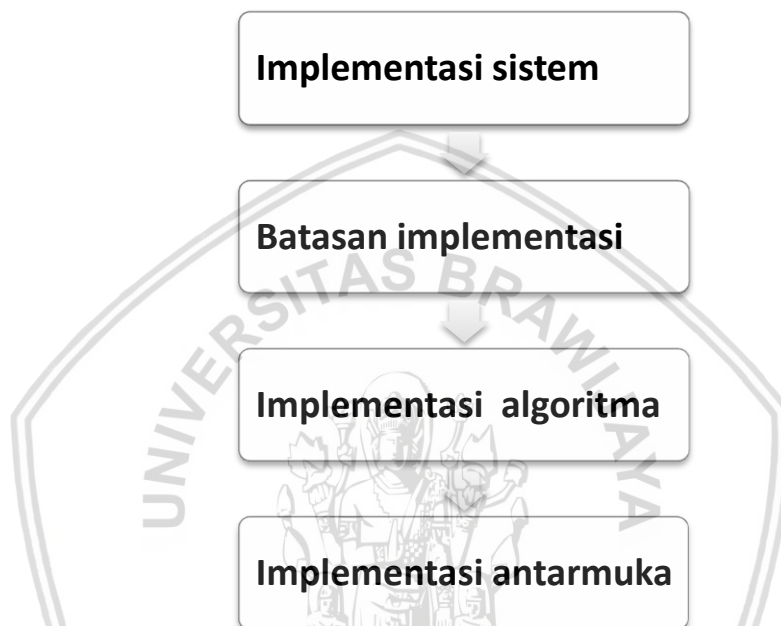
Perhitungan yang sama dilakukan untuk data uji yang lainnya yaitu data uji 2, data uji 3, data uji 4, dan data uji 5. Dari 5 data uji yang digunakan didapatkan hasil akurasi sebesar 100% dengan 5 nilai prediksi benar dan tidak ada prediksi yang salah. Hasil perhitungan dari 5 data uji dapat dilihat dalam tabel 4.12 berikut.

**Tabel 4.12 Hasil Perhitungan 5 Data Uji**

Nilai k	Data Uji	Hasil	
		Data Asli	Prediksi Sistem
3	1	1	1
	2	2	2
	3	3	3
	4	4	4
	5	5	4
Akurasi			80%

## BAB 5 IMPLEMENTASI

Bab ini membahas mengenai implementasi perangkat lunak berdasarkan hasil yang telah diperoleh dari analisis kebutuhan dan proses perancangan perangkat lunak yang dibuat. Pembahasan terdiri dari penjelasan tentang spesifikasi sistem, batasan-batasan dalam implementasi, implementasi algoritma pada program, dan implementasi antarmuka. Berikut merupakan tahapan-tahapan implementasi sistem ditunjukkan pada Gambar 5.1.



Gambar 5.1 Diagram Alir Tahap Implementasi

### 5.1 Implementasi sistem

Hasil dari proses *analisis* kebutuhan dan perancangan perangkat lunak yang telah diuraikan pada Bab 4 akan menjadi acuan untuk proses implementasi sistem. Proses implementasi sistem membutuhkan spesifikasi perangkat yang sesuai agar sistem yang dibangun dapat berfungsi sesuai dengan kebutuhan. Spesifikasi perangkat yang dibutuhkan oleh sistem terdiri dari spesifikasi perangkat keras dan spesifikasi perangkat lunak.

#### 5.1.1 Spesifikasi perangkat keras

Pengembangan Sistem untuk diagnosis hama penyakit tanaman bawang merah menggunakan perangkat komputer dengan spesifikasi perangkat keras seperti table 5.1.

Tabel 5.1 Spesifikasi Perangkat Keras

Nama komponen	Spesifikasi
Prosesor	Intel® Core™ CPU B940 @ 2.00 GHz
Memory	2.0 GB DDR3



Grafis	Intel ® HD Graphic
Hardisk	500 GB

### 5.1.2 Spesifikasi Perangkat Lunak

Pengembangan diagnosis hama penyakit tanaman bawang merah menggunakan sebuah personal komputer dengan spesifikasi perangkat lunak yang ditunjukkan pada Tabel 5.2.

**Tabel 5.2 Spesifikasi Perangkat Lunak**

Sistem operasi	Microsoft Windows 10 Home 64-bit
Bahasa pemrograman	Java
Tools pemrograman	Android Studio
Emulator	Android Emulator

## 5.2 Batasan implementasi

Batasan dalam implementasi diagnosis hama penyakit pada tanaman bawang merah adalah sebagai berikut :

1. Sistem yang dibangun berdasarkan ruang lingkup aplikasi android dengan menggunakan bahasa pemrograman java.
2. Data yang digunakan dalam sistem pakar disimpan dalam bentuk array.
3. Data yang digunakan berupa data-data gejala fisik pada tanaman bawang merah, data penyebab dari setiap gejala dan data penyakit yang menyerang tanaman bawang merah.
4. Masukan yang dilakukan oleh pengguna ke sistem berupa gejala fisik yang dialami tanaman bawang merah.
5. Keluaran dari sistem adalah salah satu dari 5 penyakit pada tanaman bawang merah.
6. Metode yang digunakan adalah *modifikasi k-nearest neighbor*.
7. Semua pengguna dapat mengakses sistem tanpa harus melakukan login.
8. Semua pengguna memiliki hak akses yang sama.
9. Semua pengguna berhak mengakses menu yang ada pada sistem.
10. Data yang digunakan pada sistem permanen dan tidak dapat diubah lagi.

## 5.3 Implementasi algoritma

Implementasi algoritma ini mengacu pada bab perancangan sub bab perancangan perangkat lunak yang mempunyai beberapa proses utama yang terbagi dalam beberapa fungsi. Pada sub bab ini hanya dicantumkan algoritma dari sebagian proses saja, sehingga tidak semua proses algoritma dicantumkan. Proses algoritma yang dicantumkan yaitu implementasi

algoritma normalisasi data, implementasi algoritma perhitungan *euclidean*, implementasi algoritma perhitungan validitas data latih dan implementasi algoritma perhitungan *weight voting*.

### 5.3.1 Implementasi algoritma perhitungan normalisasi data

Proses perhitungan normalisasi data pada metode MKNN berguna untuk menyetarakan sebaran data sehingga dapat memudahkan proses perhitungan yang menggunakan Persamaan 2.2. Proses normalisasi dilakukan pada data uji dan data latih.

```
//method untuk proses normalisasi data training
public void normalisasiDtTraining(int jmlData, int pjpgData,
int min, int max){
    this.normDtTraining = new double [jmlData][pjpgData];
    double temp;

    for (int i = 0; i < jmlData; i++){
        for (int j = 0; j < pjpgData; j++){
            if (this.dataTraining[i][j] >20) {
                temp = 90;
            } else {
                temp = 20;
            }
            this.normDtTraining[i][j] =
            (this.dataTraining[i][j]/temp)*((max+min)+min);
        }
    }

    //method untuk proses normalisasi data testing
    public void normalisasiDtTesting(int pjpgData, int min, int
    max){
        this.normDtTesting = new double [pjpgData];
        double temp;

        for (int i = 0; i < pjpgData; i++) {
            if (this.dataTesting[i] >20) {
                temp = 90;
            } else {
                temp = 20;
            }
            this.normDtTesting[i] =
            (this.dataTesting[i]/temp)*((max+min)+min);
        }
    }
}
```

**Source Code 5.1**Listing Code Proses Normalisasi Data

Penjelasan *source code* algoritma perhitungan normalisasi adalah sebagai berikut:

1. Melakukan normalisasi pada gejala 1 sampai gejala 15 di setiap data pada sistem menggunakan rumus yang telah ditentukan.

### 5.3.2 Implementasi algoritma perhitungan *euclidean*

Proses perhitungan *euclidean* ini menghitung jarak antar data latih maupun data uji yang nantinya akan digunakan dalam proses menghitung validitas data latih.

```
//method untuk menghitung euclidean
public void hitungEuclidean(int jmlData, int pjpgData){
    this.hasilEuclidean = new double [jmlData][jmlData];
    double [][] temp = new double [jmlData][jmlData];

    for (int i = 0; i < jmlData; i++) {
        for (int j = 0; j < jmlData; j++) {
            for (int k = 0; k < pjpgData; k++) {
                temp[i][j] = temp[i][j] +
                (Math.pow(this.normDtTraining[i][k]-this.normDtTraining[j][k],
                2));
            }
            this.hasilEuclidean[i][j] = Math.sqrt(temp[i][j]);
        }
    }
}
```

Source Code 5.2 Listing Code Proses Euclidean

Penjelasan *source code* algoritma perhitungan *euclidean distance* adalah sebagai berikut:

1. Menjumlahkan jarak pada setiap gejala yang kemudian hasil akhirnya *euclidean* didapat dengan cara mengakarkan hasil penjumlahan setiap gejala tersebut.

### 5.3.3 Implementasi algoritma perhitungan validitas data

Pada proses perhitungan validitas data ini dilakukan dengan membandingkan kelas pada data latih sesuai dengan nilai *k* yang diinputkan. Jika kelas yang dibandingkan sama maka akan bernilai 1 dan jika kelas yang dibandingkan tidak sama maka akan bernilai 0 sesuai dengan persamaan 2.4. Jumlah dari hasil perbandingan tersebut dibagi dengan nilai *k* yang diinputkan dan menghasilkan nilai validitas setiap data latih yang sesuai dengan persamaan 2.3.

```
//method untuk menghitung validitas
public void hitungValiditas(int jmlData, int k){
    this.nilaiK = new double [jmlData][k];
    this.sumS = new double [jmlData];
    this.tampungK = new double [jmlData][k];
    this.validitas = new double [jmlData];
    double [][] tampungEuclidean = new double [jmlData][jmlData];
    double tempNilai;

    //mengkopi nilai euclidean untuk diurutkan
    for (int i = 0; i < this.hasilEuclidean.length; i++) {
        for (int j = 0; j < this.hasilEuclidean.length; j++) {
            tampungEuclidean[i][j] = this.hasilEuclidean[i][j];
        }
    }
}
```

```

    }
}

//urutkan euclidean
boolean selesai = true;
while(selesai){
    selesai = false;
    for (int i = 1; i < tampungEuclidean.length; i++) {
        for (int j = 1; j < tampungEuclidean.length; j++) {
            if (tampungEuclidean[j-1][i] > tampungEuclidean[j][i]) {
                tempNilai = tampungEuclidean[j-1][i];
                tampungEuclidean[j-1][i] =
tampungEuclidean[j][i];
                tampungEuclidean[j][i] = tempNilai;
                selesai = true;
            }
        }
    }
}

for (int i = 0; i < tampungEuclidean.length; i++) {
    for (int j = 0; j < k; j++) {
        this.nilaiK[i][j] = tampungEuclidean[j+1][i];
    }
}

for (int i = 0; i < this.tampungK.length; i++) {
    for (int j = 0; j < k; j++) {
        if (this.nilaiK[i][0] == this.nilaiK[i][j]) {
            this.tampungK[i][j] = 1;
        } else {
            this.tampungK[i][j] = 0;
        }
        this.sumS[i] = this.sumS[i] + this.tampungK[i][j];
    }
}

double transK = k;

this.validitas[i] = (1/transK) * this.sumS[i];
}

}

//method untuk menghitung nilai euclidean data testing
public void hitungEuclideanDataTesting(int jmlDataTraining, int
pjpgData){
    double [] euclideanTemp = new double [jmlDataTraining];
    this.euclideanDtUji = new double [jmlDataTraining];

    for (int i = 0; i < this.normDtTraining.length; i++) {
        for (int j = 0; j < pjpgData; j++) {
            euclideanTemp[i] = euclideanTemp[i] +
(Math.pow(this.normDtTesting[j]-this.normDtTraining[i][j], 2));
        }
        this.euclideanDtUji[i] = Math.sqrt(euclideanTemp[i]);
    }
}

```

### Source Code 5.3 Listing Code Proses Validitas

Penjelasan *source code* algoritma perhitungan validitas data adalah sebagai berikut:

1. Memberikan nilai 0 jika kelas tidak sama dan memberikan nilai 1 jika kelas sama yang didapat dari perhitungan euclidean sebelumnya.

#### 5.3.4 Implementasi algoritma perhitungan *weight voting*

Pada proses perhitungan *weight voting* ini membahas perhitungan bobot pada data hama penyakit tanaman bawang merah, dengan cara membandingkan hasil perhitungan validitas dengan euclidean. Dari tahapan ini akan dilakukan pembobotan yang bertujuan untuk menentukan kelas pada data uji.

```
//method untuk menghitung weight voting
public void hitungWeightVoting(int k, double regulator){
    this.weightVoting = new double [this.normDtTraining.length][2];
    double [][] tempWeightVoting = new double
    [this.normDtTraining.length][2];
    double tempNilai;
    double tempKelas;

    for (int i = 0; i < this.normDtTraining.length; i++) {
        this.weightVoting[i][0] = this.validitas[i] *
        (1/(this.euclideanDtUji[i]+regulator));
        this.weightVoting[i][1] = this.kelasTraining[i];
    }

    for (int i = 0; i < this.normDtTraining.length; i++) {
        tempWeightVoting[i][0] = this.weightVoting[i][0];
        tempWeightVoting[i][1] = this.weightVoting[i][1];
    }

    //mengurutkan hasil berdasarkan nilai bobot terbesar-terkecil
    boolean selesai = true;
    while(selesai){
        selesai = false;
        for (int i = 1; i < tempWeightVoting.length; i++) {
            if (tempWeightVoting[i-1][0] < tempWeightVoting[i][0]) {
                tempNilai = tempWeightVoting[i-1][0];
                tempKelas = tempWeightVoting[i-1][1];
                tempWeightVoting[i-1][0] =
                tempWeightVoting[i][0];
                tempWeightVoting[i-1][1] =
                tempWeightVoting[i][1];
                tempWeightVoting[i][0] = tempNilai;
                tempWeightVoting[i][1] = tempKelas;
                selesai = true;
            }
        }
    }

    double [][] urutKelas = new double [k][2];
```

```

for (int i = 0; i < urutKelas.length; i++) {
    urutKelas[i][0] = tempWeightVoting[i][0];
    urutKelas[i][1] = tempWeightVoting[i][1];
}

//mengurutkan hasil berdasarkan kelas terbesar-terkecil
selesai = true;
while(selesai){
    selesai = false;
    for (int i = 1; i < urutKelas.length; i++) {
        if (urutKelas[i-1][1] < urutKelas[i][1]) {
            tempNilai = urutKelas[i-1][0];
            tempKelas = urutKelas[i-1][1];
            urutKelas[i-1][0] = urutKelas[i][0];
            urutKelas[i-1][1] = urutKelas[i][1];
            urutKelas[i][0] = tempNilai;
            urutKelas[i][1] = tempKelas;
            selesai = true;
        }
    }
}

int count = 0;
double temp = 0;
//mencari jumlah kelas
for (int i = 0; i < urutKelas.length; i++) {
    if (i == 0) {
        count++;
        temp = urutKelas[i][1];
    } else if (urutKelas[i][1] != temp) {
        temp = urutKelas[i][1];
        count++;
    }
}

//System.out.println(count);
this.hasilAkhir = new double [count][2];

int trigger = 0;
double kelas = 0;

for (int i = 0; i < urutKelas.length; i++) {
    if (i == 0) {
        this.hasilAkhir[trigger][0] = urutKelas[i][0];
        this.hasilAkhir[trigger][1] = urutKelas[i][1];
        kelas = this.hasilAkhir[i][1];
    } else {
        if (urutKelas[i][1] == kelas) {
            this.hasilAkhir[trigger][0] = this.hasilAkhir[trigger][0] +
            urutKelas[i][0];
        } else {
            trigger++;
            this.hasilAkhir[trigger][0] = urutKelas[i][0];
            this.hasilAkhir[trigger][1] = urutKelas[i][1];
            kelas = urutKelas[i][1];
        }
    }
}

```



}

**Source Code 5.4 Listing Code Proses Weight Voting**

Penjelasan *source code* algoritma perhitungan *weight voting* adalah sebagai berikut:

1. Mengurutkan nilai bobot dari yang terbesar sampai yang terkecil
2. Mengurutkan hasil berdasarkan nilai kelas terbesar sampai terkecil
3. Mencari jumlah kelas yang kemudian nilai tersebut yang menjadi acuan dalam kelas penyakit.

**5.4 Implementasi antarmuka**

Antarmuka aplikasi untuk diagnosis hama penyakit tanaman bawang merah ini digunakan oleh pengguna untuk berinteraksi dengan sistem perangkat lunak. Ada 3 antarmuka yang ditampilkan dalam aplikasi ini yaitu halaman utama, halaman pemilihan gejala dan nilai k, serta halaman hasil diagnosis.

**5.4.1 Tampilan antarmuka utama**

Tampilan antarmuka utama yang ditunjukkan pada Gambar 5.2 merupakan halaman awal ketika pengguna membuka sistem, pada halaman ini terdapat tombol atau menu yaitu Mulai. Tombol mulai tersebut digunakan untuk memulai diagnosis dengan diarahkan ke halaman pemilihan gejala dan pemilihan nilai k.



**Gambar 5.2 Tampilan Antarmuka Utama**

#### **5.4.2 Tampilan antarmuka pemilihan gejala dan nilai k**

Tampilan antarmuka pemilihan gejala dan nilai k yang ditunjukkan Gambar 5.3 merupakan halaman untuk pengguna menceklist 15 gejala pada tanaman bawang merah. Pada setiap gejala terdapat 3 pilihan tingkatan yaitu tidak ada, sedang dan tinggi, pilihan tersebut dipilih dengan gejala yang dialami oleh pengguna sesuai dengan tingkatatan gejala tersebut. Jika pengguna tidak memilih salah satu pilihan dari gejala tersebut maka sistem akan otomatis memilih pilihan yang tidak ada. Selain itu juga ada menu pemilihan nilai k yang bisa dipilih dari 1 samapai 9.

Pada halaman ini akan beralih ke halam berikutnya yaitu halam hasil diagnosis jika pengguna menekan tombol Proses.

13:25 0,00K/d 4G

### Pilih Gejala

- 1 Terdapat ulat *Spodoptera exigua* berukuran sekitar 25 mm, berwarna hijau atau coklat dengan garis tengah berwarna kuning
  - ☐ Tidak Ada
  - ☐ Sedang
  - ☒ Tinggi
- 2 Daging daun habis dan menjadi transparan
  - ☐ Tidak Ada
  - ☒ Sedang
  - ☐ Tinggi
- 3 Terdapat ulat yang menyerang tanaman dengan memakan daun bagian dalam
  - ☐ Tidak Ada
  - ☐ Sedang
  - ☒ Tinggi
- 4 Terdapat Lalat Pengorok, gejala
  - ☐ Tidak Ada
  - ☐ Tinggi

Nilai K : 1

PROSES

**Gambar 5.3 Tampilan Antarmuka Pemilihan Gejala dan Nilai k**

#### 5.4.3 Tampilan antarmuka hasil diagnosis

Tampilan antarmuka hasil diagnosis yang ditunjukkan Gambar 5.4 merupakan halaman untuk mengetahui hasil diagnosis yang didapat dari masukan pengguna pada halaman pemilihan gejala dan nilai k. Hasil yang dimaksud yaitu jenis hama penyakit yang dialami dan saran penanganan yang harus dilakukan.



**Gambar 5.4 Tampilan Antarmuka Hasil Diagnosis**

## BAB 6 PENGUJIAN DAN ANALISIS

Pada bab ini akan dibahas mengenai pengujian algoritma yang terdiri dari skenario pengujian dan analisis hasil pengujian diagnosis hama penyakit pada tanaman bawang merah. Pengujian yang dilakukan adalah pengujian akurasi.

### 6.1 Pengujian pengaruh nilai k

Pengujian pengaruh nilai k dilakukan untuk mengetahui apakah nilai k berpengaruh terhadap hasil akurasi sistem jika nilai k tersebut diubah. Pengujian ini dilakukan dengan cara mengubah nilai k dari k=1 sampai k=10 kemudian masing-masing nilai k tersebut dicoba pada saat data latih 20, data latih 30 dan data latih 44. Hasil akurasi dari masing-masing nilai k pada data latih yang berbeda dirata-rata dan kemudian hasil tersebut dibandingkan untuk dicari nilai k yang menghasilkan akurasi sistem tertinggi.

#### 6.1.1 Skenario Pengujian Pengaruh Nilai k

Prosedur pengujian pengaruh nilai k dilakukan dengan cara menghitung nilai akurasi pada k=1 sampai k=10. Setiap nilai k akan dilakukan uji coba sebanyak 3 kali dan didapatkan nilai akurasi rata-rata. Jumlah data latih yang digunakan terdapat 3 macam yaitu data latih 20, data latih 30 dan data latih 44.

#### 6.1.2 Analisis Hasil Skenario Pengujian Pengaruh Nilai k

Pada pengujian pengaruh nilai k yang telah dilakukan, didapatkan rata-rata nilai akurasi yang berbeda-beda. Nilai akurasi rata-rata tertinggi sebesar 85.5% pada saat nilai k=4, sedangkan nilai akurasi rata-rata terendah yang dihasilkan sebesar 78.6% pada saat nilai k=10. Hasil pengujian pengaruh nilai k ditunjukkan pada Tabel 6.16. Sedangkan untuk grafik pengaruh nilai k ditunjukkan pada gambar 6.1.

**Tabel 6.1 Hasil Pengujian Pengaruh Nilai k**

Nilai k	Akurasi Pengujian Untuk Data Latih (%)			Rata – Rata Akurasi (%)
	20	30	44	
1	76	80	83	79.9
2	76	83	83	80.6
3	80	83	90	84.3
4	83	83	90	85.5
5	83	86	86	85.0
6	80	86	86	84.0
7	80	86	86	84.0

8	76	83	83	80.6
9	76	80	83	79.6
10	73	80	83	78.6

**Gambar 6.1 Grafik pengaruh nilai k terhadap rata-rata akurasi**

## 6.2 Pengujian akurasi sistem

Pengujian ini dilakukan dengan menguji tingkat keakuratan atau kesesuaian dari data testing yang didapatkan oleh pakar dengan hasil output dari sistem. Data testing yang didapatkan sebanyak 30 dan data latih yang digunakan dalam sistem sebanyak 44, keseluruhan data latih didapatkan dari pakar. Kemudian akan dilakukan percobaan dengan masukan sesuai data uji, kemudian nilai akurasi akan didapatkan dari perhitungan menggunakan Persamaan 2.7.

Untuk mendapatkan nilai akurasi akan dilakukan uji kecocokan antara keluaran sistem dengan data uji yang didapatkan dari pakar, dimana data uji dari pakar ditunjukkan oleh Tabel 6.2 dan hasil uji sistem ditunjukkan oleh Tabel 6.3.

**Tabel 6.2 Data Uji Dari Pakar**

NO	G1	G2	G3	G4	G5	G6	G7	G8	G9	G10	G11	G12	G13	G14	G15	KELAS
1	90	10	20	0	0	0	45	0	0	10	0	0	0	10	0	1
2	90	10	10	10	0	0	0	0	0	0	20	0	0	0	0	1
3	45	20	20	0	0	0	0	0	10	0	0	0	0	0	0	1
4	45	20	10	0	10	0	0	0	0	0	0	0	0	0	10	1
5	45	10	20	0	0	0	0	0	0	0	10	0	10	0	0	1
6	45	0	10	0	20	90	0	10	0	0	0	0	0	0	20	1
7	0	0	0	20	10	90	0	0	0	0	0	0	0	10	0	2
8	0	0	0	10	10	90	0	0	10	0	10	0	0	0	0	2
9	0	10	0	20	20	45	0	0	0	0	0	0	0	20	0	2
10	0	0	10	10	20	45	0	0	0	0	0	0	0	0	0	2
11	0	0	0	20	10	45	0	0	0	10	0	0	0	0	0	2
12	0	0	0	20	0	90	0	10	0	10	10	0	0	0	0	2
13	0	20	0	0	0	0	90	10	20	10	0	0	0	10	0	3
14	0	0	0	0	10	0	90	10	10	20	0	0	0	0	0	3
15	0	0	10	0	0	0	90	10	10	10	0	0	0	10	0	3

16	0	0	0	0	10	0	45	20	20	20	0	0	0	0	10	3
17	0	10	0	0	0	0	45	20	20	10	0	0	0	0	0	3
18	0	0	0	0	0	0	0	10	0	10	20	90	0	0	0	3
19	0	0	0	0	0	10	0	0	0	0	20	90	0	10	0	4
20	0	0	0	10	0	0	0	0	0	0	10	90	0	0	0	4
21	0	0	20	0	0	0	0	0	0	0	20	45	0	20	0	4
22	0	0	0	0	10	0	0	0	0	10	10	45	0	0	0	4
23	0	20	0	0	0	10	0	0	0	0	0	90	90	0	0	4
24	0	0	0	0	0	0	0	10	0	0	0	90	0	10	20	4
25	0	0	0	0	20	0	0	0	0	0	0	0	90	10	20	5
26	0	10	0	10	0	0	0	0	0	0	20	0	90	10	10	5
27	0	0	0	0	0	0	0	0	0	10	0	0	45	20	20	5
28	0	0	20	0	0	0	0	0	0	0	0	0	45	20	10	5
29	0	0	0	0	0	10	0	0	10	0	0	0	45	10	20	5
30	0	10	0	0	10	0	0	0	0	0	10	90	0	0	20	5

Tabel 6.3 Hasil Uji Keluaran Sistem

NO	G1	G2	G3	G4	G5	G6	G7	G8	G9	G10	G11	G12	G13	G14	G15	KELAS
1	90	10	20	0	0	0	45	0	0	10	0	0	0	10	0	1
2	90	10	10	10	0	0	0	0	0	0	20	0	0	0	0	1
3	45	20	20	0	0	0	0	0	10	0	0	0	0	0	0	1
4	45	20	10	0	10	0	0	0	0	0	0	0	0	0	10	1
5	45	10	20	0	0	0	0	0	0	0	10	0	10	0	0	1
6	45	0	10	0	20	90	0	10	0	0	0	0	0	0	20	2
7	0	0	0	20	10	90	0	0	0	0	0	0	0	10	0	2
8	0	0	0	10	10	90	0	0	10	0	10	0	0	0	0	2
9	0	10	0	20	20	45	0	0	0	0	0	0	0	20	0	2
10	0	0	10	10	20	45	0	0	0	0	0	0	0	0	0	2
11	0	0	0	20	10	45	0	0	0	10	0	0	0	0	0	2
12	0	0	0	20	0	90	0	10	0	10	10	0	0	0	0	2



13	0	20	0	0	0	0	90	10	20	10	0	0	0	10	0	3
14	0	0	0	0	10	0	90	10	10	20	0	0	0	0	0	3
15	0	0	10	0	0	0	90	10	10	10	0	0	0	10	0	3
16	0	0	0	0	10	0	45	20	20	20	0	0	0	0	10	3
17	0	10	0	0	0	0	45	20	20	10	0	0	0	0	0	3
18	0	0	0	0	0	0	0	10	0	10	20	90	0	0	0	4
19	0	0	0	0	0	10	0	0	0	0	20	90	0	10	0	4
20	0	0	0	10	0	0	0	0	0	0	10	90	0	0	0	4
21	0	0	20	0	0	0	0	0	0	0	20	45	0	20	0	4
22	0	0	0	0	10	0	0	0	0	10	10	45	0	0	0	4
23	0	20	0	0	0	10	0	0	0	0	0	90	90	0	0	5
24	0	0	0	0	0	0	0	10	0	0	0	90	0	10	20	5
25	0	0	0	0	20	0	0	0	0	0	0	0	90	10	20	5
26	0	10	0	10	0	0	0	0	0	0	20	0	90	10	10	5
27	0	0	0	0	0	0	0	0	0	10	0	0	45	20	20	5
28	0	0	20	0	0	0	0	0	0	0	0	0	45	20	10	5
29	0	0	0	0	0	10	0	0	10	0	0	0	45	10	20	5
30	0	10	0	0	10	0	0	0	0	0	10	90	0	0	20	4

Dari percobaan yang telah dilakukan sebanyak 30 data uji didapatkan hasil keluaran sistem yang sesuai sebanyak 25 dimana ketidak sesuaian keluaran dari sistem dengan hasil uji dari pakar terdapat pada data uji ke-6, ke-18, ke-23, ke-24 dan ke-30. Sehingga didapatkan nilai akurasi menggunakan persamaa 2-7 sebagai berikut:

$$Akurasi = \frac{25}{30} \times 100$$

Maka Akurasi = 83,33%

### 6.3 Analisis akurasi

Dalam penelitian ini terdapat dua pengujian yaitu pengujian manualisasi dan pengujian sistem. Pada pengujian manualisasi peneliti hanya menggunakan data latih sebanyak 10 dan data uji sebanyak 5 yang diambil acak dari 30 data uji pakar, keakurasiannya yaitu 80%. Sedangkan dalam

pengujian sistem yang menggunakan 44 data latih dan 30 data uji dari pakar, tingkat keakurasiannya lebih tinggi yaitu 83,33%. Dari dua pengujian tersebut dapat diambil kesimpulan bahwa jumlah data latih sangat mempengaruhi tingkat keakurasian.



## BAB 7 PENUTUP

### 7.1 Kesimpulan

Berdasarkan hasil pengujian dan analisis hasil penelitian ini, maka dapat diambil beberapa kesimpulan sebagai berikut :

1. Metode *modifikasi k-nearest neighbor* berhasil diimplentasi dalam diagnosis hama penyakit pada tanaman bawang merah dalam bentuk perangkat lunak dengan fungsi melakukan diagnosis hama penyakit pada tanaman bawang merah. Beserta penanganan yang harus dilakukan oleh pengguna terhadap tanaman bawang merah yang terkena penyakit.
2. Hasil oppengujiandari diagnosis penyakit tanaman bawang merah menggunakan metode *Modified K-Nearest Neighborm* memperoleh tingkat akurasi sebesar 83%. Jadi metode ini cukup akurat untuk di implementasikan dalam permasalahan diagnosis hama penyakit tanaman bawang merah.

### 7.2 Saran

Berdasarkan penelitian yang telah dilakukan, maka saran untuk penelitian selanjutnya adalah sebagai berikut:

1. Diagnosis hama penyakit tanaman bawang merah ini menghasilkan akurasi yang tinggi tetapi memiliki kelemahan yaitu jika gejala yang dimasukkan hanya terdiri dari satu gejala dan gejanya bukan gejala utama suatu hama penyakit tanaman bawang merah atau bisa dibilang satu gejala tersebut dapat mewakili lebih dari satu penyakit maka akan terjadi kemungkinan salah diagnosis karena keluaran dari sistem hanya berdasarkan peluang pada data latih, oleh karena itu akan lebih baik jika metode *modifikasi k-nearest neighbor* digabung atau dioptimasi dengan suatu algoritma yang lain untuk menanggulangi kekurangan tersebut.
2. Diagnosis hama penyakit tanaman bawang merah ini menghasilkan akurasi yang cukup tinggi tetapi akan lebih bias tinggi lagi jika didapatkan data lating yang lebih banyak. Untuk saat ini dalam penelitian ini menggunakan 15 gejala dari 5 hama penyakit yang ditimbulkan oleh bawang merah, kemungkinan besar di masa akan datang akan terdapat lagi gejala-gejala baru yang akan bermunculan bahkan terdapat hama penyakit baru lagi. Jadi saran untuk penelitian berikutnya yang berkaitan dengan hama penyakit tanaman bawang merah bisa lebih lagi mencari gejala-gejala atau penyakit baru yang belum terdapat pada penelitian ini.

## DAFTAR PUSTAKA

- Aktivita, R. (2012). *Pengenalan Iris Mata Menggunakan Algoritma K-Nearest Neighbor Berbasis Histogram*. S1. Institut Pertanian Bogor.
- Baswarsiaty, E., Korlina, K. B., Andri, L., Rosmahani, & B., I. *Teknologi Usaha Tani Bawang Merah Spesifikasi Jawa Timur*.
- Han, J., & Kamber, M. (2006). *Data Mining : Concepts and Techniques*. San Fransisco: Morgan Kauffman Publisher.
- Kantardzic, M. (2003). *Data Mining: Concepts, Models, Methods and Algorithm*. New York: John Wiley & Sons.
- Kusnawi. (2007). *Pengantar Solusi Data Mining*. Seminar Nasional Teknologi 2007 (SNT 2007), Issue ISSN : 1978 – 9777.
- Larose, D. T. (2005). *Discovering Knowledge in Data*. New Jersey: John Wiley & Sons, Inc.
- Moertini, V. (2002). *Data mining Sebagai Solusi Bisnis*. Volume 7.
- Mutrofin, S., Izzah, A., Kurniawardhani, A., & Masrur, M. *Optimization Techniques Modified k Nearest Neighbor Classification Using Genetic*. Fakultas Teknik, Unipdu.
- P.B, M. (2015). *Diagnosis Penyakit Tanaman Tomat menggunakan Algoritma Modified K-Nearest Neighbor(MKNN)*. Malang: Universitas Brawijaya.
- Parvin, H., Alizadeh, H., & Minati, B. (2010). *A Modification on K-Nearest Neighbor*. Global Journal of Computer Science and Technology.
- Pertanian, D. (2009). *Statistik pertanian 2009, Pusat Data dan Informasi Pertanian*. Jakarta: Departemen Pertanian.
- Pracaya. (2009). *Hama dan Penyakit Tanaman*.
- Pramudiono, I. (2003). *Pengantar Data mining: Menambang Permata Pengetahuan di Gunung Data*. IlmuKomputer.com.
- Sulistiyandari. (2012). *Penerapan Algoritma Modified K-Nearest Neighbor(MKNN) untuk Mengklasifikasikan Letak Protein pada Bakteri Pengaruh Pb<sup>2+</sup> terhadap Aktivitas Pektinase dari Bacillus firmus E-Coli*. S1. Universitas Brawijaya.
- Udiarto, Setiawati, & Suryaningsih. (2005). *Pengenalan Hama dan Penyakit pada Tanaman Bawang Merah dan Pengendaliannya*. Bandung: Panduan Teknis PTT Bawang Merah No 2. Balai Penelitian Tanaman Sayuran.

Wafiyah, F., Hidayat, N., & Perdana, R. S. (2017). *Implementasi Algoritma Modified K-Nearest Neighbor (MKNN) untuk Klasifikasi Penyakit Demam*. Malang: Fakultas Ilmu Komputer, Universitas Brawijaya.

Yu, C. (2010). *How KNN Works*. Indiana University.

Zainuddin, S., Hidayat, N., & Soebroto, A. A. (Penerapan Algoritma Modified K-Nearest Neighbour (M-KNN) pada Pengklasifikasian Penyakit Tanaman Kedelai). *Penerapan Algoritma Modifie*. Malang: S1. Universitas Brawijaya.



## LAMPIRAN AHASIL WAWANCARA DENGAN PAKAR

### HASIL WAWANCARA DENGAN PAKAR

Tempat : Balai Pengkajian Teknologi Pertanian (BPTP) Jawa Timur

Tanggal : 28 November 2017

Nama Pakar : Prof. Dr. Ir. Moh. Cholil Mahfud, M.S.

No	Pertanyaan	Jawaban Pakar
1	Apakah di Balai Pengkajian Teknologi Pertanian (BPTP) Jawa Timur ini terdapat budidaya tanaman bawang merah?	Ya, di Balai Pengkajian Teknologi Pertanian (BPTP) Jawa Timur ini ada budidaya bawang merah.
2	Apakah tanaman bawang merah di BPTP memiliki beberapa kendala seperti serangan hama penyakit bawang merah?	Ya, disini terdapat beberapa hama penyakit bawang merah
3	Hama penyakit apa saja yang menyerang pada tanaman bawang merah?	Ada 5 hama penyakit utama pada bawang merah yaitu Hama Ulat Bawang, Hama Lalat Pengorok Daun, Penyakit Moler, Penyakit Trotol, dan Penyakit Antraknose.
4	Saya akan meneliti tentang hama penyakit tanaman bawang merah untuk diaplikasikan kedalam sebuah sistem pakar yang kemudian akan tercipta sebuah aplikasi diagnosis hama penyakit bawang merah berbasis android. Apakah saya bisa menggunakan data hama penyakit yang terdapat di BPTP?	Bisa, disini terdapat banyak data hama penyakit bawang merah, sebelumnya juga terdapat penelitian dari banyak mahasiswa yang menggunakan data dari sini juga.
5	Apa sajakah parameter yang digunakan untuk mendiagnosis hama penyakit bawang merah yang dilakukan oleh pakar?	Parameter yang digunakan yaitu berdasarkan dari gejala-gejala yang dialami dari tanaman bawang merah.

Mengetahui,

Prof. Dr. Ir. Moh. Cholil Mahfud, M.S.



## LAMPIRAN B NILAI PEMBOBOTAN GEJALA HAMA PENYAKIT BAWANG MERAH

**NILAI PEMBOBOTAN GEJALA HAMA PENYAKIT BAWANG MERAH**

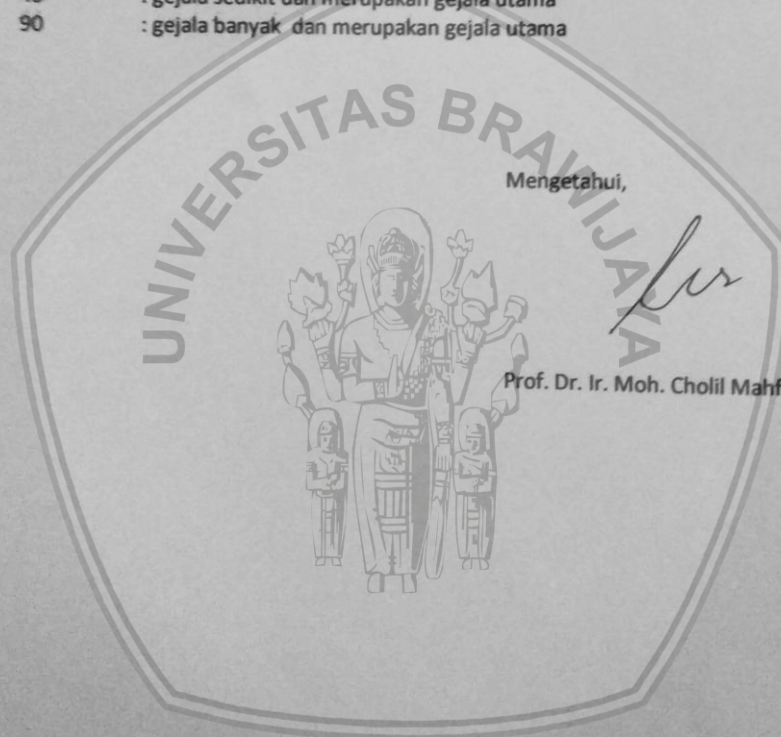
No	Kode	Gejala	Opsl	Nilai
1	G1	Terdapat Ulat Spodoptera exigua berukuran sekitar 25 mm, berwarna hijau atau coklat dengan garis tengah berwarna kuning.	tidak ada	0
			sedikit	45
			banyak	90
2	G2	Daging daun habis dan menjadi transparan.	tidak ada	0
			sedikit	10
			banyak	20
3	G3	Terdapat ulat yang menyerang tanaman dengan memakan daun bagian dalam.	tidak ada	0
			sedikit	10
			banyak	20
4	G4	Terdapat Lalat Pengorok, gejala serangan mulai pada umur 15 hari setelah tumbuh hingga menjelang panen.	tidak ada	0
			sedikit	10
			banyak	20
5	G5	Daun berwarna coklat seperti terbakar dan masuk ke dalam umbi bawang.	tidak ada	0
			sedikit	10
			banyak	20
6	G6	Hampir seluruh helaian daun penuh kerokan.	tidak ada	0
			sedikit	45
			banyak	90
7	G7	Daun bawang merah menguning dan terpelintir layu (moler).	tidak ada	0
			sedikit	45
			banyak	90
8	G8	Tanaman mudah tercabut karena pertumbuhan akar terganggu dan membusuk.	tidak ada	0
			sedikit	10
			banyak	20
9	G9	Apabila umbi lapis dipotong membujur akan terlihat adanya pembusukan berawal dari dasar umbi meluas ke atas dan ke samping.	tidak ada	0
			sedikit	10
			banyak	20
10	G10	Tanaman kurus kekuningan dan busuk bagian pangkal.	tidak ada	0
			sedikit	10
			banyak	20
11	G11	Ujung daun mengering dan daun patah.	tidak ada	0
			sedikit	10
			banyak	20
12	G12	Daun timbul bercak berukuran kecil, berwarna putih dengan pusat berwarna ungu.	tidak ada	0
			sedikit	45
			banyak	90
13	G13	Terbentuk Lekukan ke dalam, berlubang dan	tidak ada	0



		patah.	sedikit	45
			banyak	90
14	G14	Terdapat bercak berwarna putih pada daun.	tidak ada	0
			sedikit	10
			banyak	20
15	G15	Tanaman akan mati dengan cepat dan mendadak.	tidak ada	0
			sedikit	10
			banyak	20

Keterangan :

- 0 : tidak ada gejala
- 10 : gejala sedikit tapi bukan gejala utama
- 20 : gejala banyak tapi bukan gejala utama
- 45 : gejala sedikit dan merupakan gejala utama
- 90 : gejala banyak dan merupakan gejala utama



Mengetahui,

Prof. Dr. Ir. Moh. Cholil Mahfud, M.S.

## LAMPIRAN C DATA LATIH SISTEM

DATA LATIH SISTEM

No	G1	G2	G3	G4	G5	G6	G7	G8	G9	G10	G11	G12	G13	G14	G15	kelas
1	90	20	20	0	20	0	45	0	0	20	0	45	0	20	0	1
2	90	20	10	0	0	45	0	10	0	0	20	0	45	0	0	1
3	90	10	20	0	20	0	45	0	20	0	10	0	0	0	20	1
4	90	10	10	10	0	90	0	0	10	0	0	0	45	0	0	1
5	45	20	20	0	20	0	0	10	0	10	0	45	0	0	10	1
6	45	20	10	0	0	45	0	10	0	20	0	0	0	0	20	1
7	45	10	20	20	0	0	45	0	10	0	10	0	0	0	0	1
8	45	10	10	0	10	0	0	10	0	0	10	0	45	0	10	1
9	0	10	0	20	20	90	0	0	10	0	20	0	0	0	0	2
10	0	20	0	10	20	90	0	10	0	10	0	0	0	0	0	2
11	0	10	0	20	10	90	0	0	10	0	10	0	0	20	0	2
12	0	20	0	10	10	90	0	10	0	0	0	0	45	0	0	2
13	0	0	10	20	20	45	0	0	20	0	10	45	0	0	20	2
14	45	0	0	10	20	45	0	0	0	10	0	0	0	0	0	2
15	0	10	0	20	10	45	0	0	10	0	0	0	45	0	0	2
16	0	0	20	10	10	45	0	0	0	0	10	0	0	0	0	2
17	0	0	10	0	0	0	90	20	20	20	0	0	0	0	10	3
18	0	20	0	0	0	0	90	20	20	10	0	0	45	20	0	3
19	0	0	0	10	0	0	90	20	10	20	0	0	0	10	0	3
20	45	0	0	0	10	45	90	20	10	10	10	0	0	0	0	3
21	0	0	0	20	0	0	90	10	20	20	0	0	0	20	0	3
22	45	0	0	0	0	0	90	10	20	10	0	45	0	0	0	3
23	0	0	10	0	0	45	90	10	10	20	0	0	0	10	0	3
24	0	0	0	0	0	0	90	10	10	10	10	0	0	0	10	3
25	0	20	0	0	0	0	45	20	20	20	0	0	45	0	0	3
26	0	0	0	10	20	0	45	20	20	10	0	0	0	0	20	3
27	45	0	0	0	0	0	45	20	10	20	0	45	0	0	0	3
28	0	0	20	0	0	0	45	20	10	10	0	0	0	0	10	3
29	0	0	0	0	0	45	45	10	20	20	0	0	0	0	0	3
30	0	0	0	0	0	0	45	10	20	10	0	0	0	0	0	3
31	0	20	0	0	0	0	45	10	10	20	0	0	0	20	0	3
32	45	0	0	20	0	0	45	10	10	10	0	0	45	0	0	3

33	0	0	0	0	0	0	0	0	0	0	0	20	90	0	0	10	4
34	0	20	0	10	0	0	0	0	10	0	10	90	0	0	0	0	4
35	0	0	0	20	0	45	0	0	0	0	20	45	0	10	0	0	4
36	0	10	0	0	0	0	0	10	20	0	10	45	0	0	0	0	4
37	0	0	0	10	20	0	45	0	0	0	0	0	90	20	20	5	
38	45	0	20	0	0	0	0	10	0	0	20	0	90	20	10	5	
39	0	0	10	0	0	0	0	20	0	10	0	45	90	10	20	5	
40	0	20	0	0	0	45	0	0	20	0	0	0	90	10	10	5	
41	0	0	0	10	20	0	0	0	0	10	10	0	45	20	20	5	
42	0	0	10	0	0	0	45	0	0	0	20	0	45	20	10	5	
43	90	10	0	0	0	0	0	10	0	0	0	0	45	10	20	5	
44	0	0	0	20	0	0	0	0	20	0	0	0	45	10	10	5	

Keterangan kelas hama penyakit :

- 1 : Hama Ulat Bawang
- 2 : Hama Lalat Pengorok Daun
- 3 : Penyakit Moler
- 4 : Penyakit Trotol
- 5 : Penyakit Antraknose

Mengetahui,

Prof. Dr. Ir. Moh. Cholil Mahfud, M.S.



## LAMPIRAN DDATA KASUS UJI

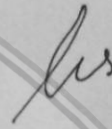
DATA KASUS UJI

NO	G1	G2	G3	G4	G5	G6	G7	G8	G9	G10	G11	G12	G13	G14	G15	KELAS
1	90	10	20	0	0	0	45	0	0	10	0	0	0	10	0	1
2	90	10	10	10	0	0	0	0	0	0	20	0	0	0	0	1
3	45	20	20	0	0	0	0	0	10	0	0	0	0	0	0	1
4	45	20	10	0	10	0	0	0	0	0	0	0	0	0	10	1
5	45	10	20	0	0	0	0	0	0	0	10	0	10	0	0	1
6	45	0	10	0	20	90	0	10	0	0	0	0	0	0	20	1
7	0	0	0	20	10	90	0	0	0	0	0	0	0	10	0	2
8	0	0	0	10	10	90	0	0	10	0	10	0	0	0	0	2
9	0	10	0	20	20	45	0	0	0	0	0	0	0	20	0	2
10	0	0	10	10	20	45	0	0	0	0	0	0	0	0	0	2
11	0	0	0	20	10	45	0	0	0	10	0	0	0	0	0	2
12	0	0	0	20	0	90	0	10	0	10	10	0	0	0	0	2
13	0	20	0	0	0	0	90	10	20	10	0	0	0	10	0	3
14	0	0	0	0	10	0	90	10	10	20	0	0	0	0	0	3
15	0	0	10	0	0	0	90	10	10	10	0	0	0	10	0	3
16	0	0	0	0	10	0	45	20	20	20	0	0	0	0	10	3
17	0	10	0	0	0	0	45	20	20	10	0	0	0	0	0	3
18	0	0	0	0	0	0	0	10	0	10	20	90	0	0	0	3
19	0	0	0	0	0	10	0	0	0	0	20	90	0	10	0	4
20	0	0	0	10	0	0	0	0	0	0	10	90	0	0	0	4
21	0	0	20	0	0	0	0	0	0	0	20	45	0	20	0	4
22	0	0	0	0	10	0	0	0	0	10	10	45	0	0	0	4
23	0	20	0	0	0	10	0	0	0	0	0	90	90	0	0	4
24	0	0	0	0	0	0	0	10	0	0	0	90	0	10	20	4
25	0	0	0	0	20	0	0	0	0	0	0	0	90	10	20	5
26	0	10	0	10	0	0	0	0	0	0	20	0	90	10	10	5
27	0	0	0	0	0	0	0	0	0	10	0	0	45	20	20	5
28	0	0	20	0	0	0	0	0	0	0	0	0	45	20	10	5
29	0	0	0	0	0	10	0	0	10	0	0	0	45	10	20	5
30	0	10	0	0	10	0	0	0	0	0	10	90	0	0	20	5

Keterangan kelas hama penyakit :

- 1 : Hama Ulat Bawang
- 2 : Hama Lalat Pengorok Daun
- 3 : Penyakit Moler
- 4 : Penyakit Trotol
- 5 : Penyakit Antraknose

Mengetahui,



Prof. Dr. Ir. Moh. Cholil Mahfud, M.S.

